



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“Diseño de una red de dispositivos telemédicos para el monitoreo de
infantes con síntomas de enfermedades respiratorias”**

INFORME DE PROYECTO INTEGRADOR

Previa a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

FRAY RUDYARB COBEÑA ANDRADE

JOHN JAIRO UGALDE CEDEÑO

GUAYAQUIL – ECUADOR

AÑO: 2018

DEDICATORIA

Este trabajo va dedicado para mi mamá como pilar fundamental en mi vida, a mis tíos y en especial a mis abuelos que jugaron un papel importante para mi desarrollo como persona. A mi padre que me enseñó cómo funciona la vida afuera de la universidad.

Fray Rudyarb Cobeña Andrade

El presente proyecto lo dedico a Dios como principal apoyo de mi familia, mis abuelos y en especial a mis padres que son el pilar de mi vida ya que sin su apoyo y dedicación esto no sería posible y a cada uno de los que intervinieron en mi formación profesional.

John Jairo Ugalde Cedeño

AGRADECIMIENTOS

Mis más sinceros agradecimientos a mis profesores por ayudarme a formarme como profesional, a nuestro tutor y profesora de materia integradora por el constante apoyo. A mis compañeros y amigos del club de ROBOTA por ser como una familia para mí, a mi novia por ser un gran apoyo para mí en esta etapa, a mis tutores y consejeros de ROBOTA por guiarme también en este gran camino que fue la universidad. Le quiero agradecer a mi mami Gilma y mi papi Manuel por cuidarme hasta los últimos días de sus vidas.

Fray Rudyarb Cobeña Andrade

A Dios, mis padres y familiares, a nuestra profesora de la materia integradora que ha sido un gran apoyo en cada uno de los avances y resultados obtenidos, siendo bases importantes para realizar este trabajo.

John Jairo Ugalde Cedeño

DECLARACIÓN EXPRESA

"Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *Fray Rudyarb Cobeña Andrade* y *John Jairo Ugalde Cedeño* 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"

Fray Rudyarb Cobeña Andrade

John Jairo Ugalde Cedeño

EVALUADORES

María Antonieta Alvarez Villanueva

PROFESORA DE LA MATERIA

José Félix Moncayo Rea

PROFESOR TUTOR

RESUMEN

En este trabajo se analiza los problemas relacionados con las enfermedades respiratorias que existen en el país, tales como la neumonía, bronquitis, influenza y demás. Cabe destacar que el análisis se realizó para una población de infantes de 0 a 4 años que representan uno de los sectores más vulnerables del país. Nuestro objetivo es diseñar un prototipo de red que permite enlazar diferentes equipos telemédicos, utilizando tecnologías IoT (por sus siglas en inglés internet de las cosas), para el monitoreo en tiempo real de pacientes con enfermedades respiratorias.

Se necesita un dispositivo que permita monitorear los signos vitales de los pacientes que se encuentran en una sala de hospital, dado que existe déficit de personal médico. Es por esto que se propone una solución desde el punto de vista del monitoreo de los pacientes.

Para realizar las pruebas de la red se utilizó un termómetro infrarrojo digital sin contacto y un oxímetro de pulso y frecuencia cardiaca, además para el procesamiento de los datos se usó una tarjeta de adquisición (NodeMCU ESP8266N), misma que se comunica a través de un enlace inalámbrico con una base de datos en tiempo real, para luego ser leídos por la aplicación móvil, la cual es la encargada de mostrar la información.

Se logró realizar el prototipo de la red de equipos telemédicos y la aplicación móvil. Por medio de pruebas se determinó que la información que observamos en la aplicación es en tiempo real. Al final se pudo observar el tiempo de respuesta de la red en términos de envío de información, conocer las opiniones del personal médico respecto a al prototipo y ver como los datos eran afectados por diferentes circunstancias como la interferencia co-canal y pérdidas en interiores.

Palabras Clave: Déficit, Médicos, IoT, Diseño, Red

ABSTRACT

This paper analyzes the problems related to respiratory diseases that exist in the country, such as pneumonia, bronchitis, influenza and others. It should be noted that the analysis was carried out for a population of infants from 0 to 4 years old who represent one of the most vulnerable sectors of the country. Our goal is to design a prototype network that allows linking different telemedicine devices, using IoT (for its acronym in English, internet of things), for the real-time monitoring of patients with respiratory diseases.

A device is needed to monitor the vital signs of patients in a hospital ward, given that there is a shortage of medical personnel. That is why a solution is proposed from the point of view of patient monitoring.

A non-contact digital infrared thermometer and a heart rate and pulse oximeter were used to perform the network tests, and an acquisition card (NodeMCU ESP8266N) was used to process the data. wireless link with a database in real time, to be read by the mobile application, which is responsible for displaying the information.

The prototype of the telemedical equipment network and the mobile application were achieved. Through tests it was determined that the information we observe in the application is in real time. In the end it was possible to observe the response time of the network in terms of sending information, to know the opinions of the medical personnel regarding the prototype and to see how the data were affected by different circumstances such as co-channel interference and internal losses.

Keywords: Deficit, Medical, IoT, Design, Network

ÍNDICE GENERAL

RESUMEN	I
ABSTRACT	II
ÍNDICE GENERAL.....	III
ABREVIATURAS	V
SIMBOLOGÍA.....	VI
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	VIII
CAPÍTULO 1	1
1. Análisis de componentes tecnológicos y problemática.	1
1.1. Planteamiento del problema	4
1.2. Objetivos.....	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Específicos	5
1.3. Justificación del problema.....	5
1.4. Marco Teórico.....	6
1.4.1. Sensores	6
1.4.2. Tarjeta controladora Hiletgo.	7
1.4.3. Tecnología Wi-Fi	8
1.4.4. Plataforma IoT.....	11
Capítulo 2.....	14
2. Sistema de sensado y manejo de la información	14
2.1. Topología de la red.....	15
2.2. Manejo de la información	16
2.3.1. NodeMCU	16
2.3.2. Google Firebase (Base de datos en tiempo real)	20
2.3.3. Visualización (Aplicación móvil).....	22
CAPÍTULO 3.....	26
3. El monitoreo en la nube y porqué usarlo.....	26
3.1. Fisiología de la red	26
3.2. Cálculo de pérdidas en interiores.....	27
3.3. Desempeño de la red.....	29

3.3.1.	Tiempo de transmisión de datos.....	29
3.3.2.	Pérdidas de paquetes	30
3.3.3.	Interferencia	32
3.4.	Costo y Mantenimiento	35
3.4.1.	Costo de la Implementación	35
3.4.2.	Mantenimiento de la solución propuesta	36
3.5.	Retroalimentación.....	36
Capítulo 4.....		40
4.	Conclusiones y Recomendaciones	40
4.1.	Conclusiones	40
4.2.	Recomendaciones	41
BIBLIOGRAFÍA		42
APÉNDICES		44

ABREVIATURAS

IoT	Internet Of Things
OMS	Organización Mundial de la Salud
INEC	Instituto Nacional de Estadísticas y Censos
NTC	Negative Temperature Coefficient
GPIO	General Purpose Input/Output
CPU	Central Processing Unit
IEEE	Institute of Electrical and Electronics Engineers
BSS	Basic Service Set
BSA	Basic Service Area
ESS	Extended Service Set
AP	Access point
DS	Distribution System
MSDU	Unidades de Dato de Servicios MAC
FDDI	Interfaz de Datos Distribuidos por Fibra
GFSK	Desplazamiento Gaussianos de Frecuencia
DSSSS	Espectro Ensanchado por Secuencia Directa
BDPSK	Modulación por Desplazamiento de Fase Binaria Diferencial
PPM	Pulse-Position Modulation
GPS	Sistema de Posicionamiento Global
RFID	Identificación por Radiofrecuencia
Wi-Fi	Wireless Fidelity
JSON	Notación de Objetos de JavaScript

SIMBOLOGÍA

GHz	Gigahertz
Nm	Nanómetros
Ms	Milisegundos
Mbit/s	Megabit por segundo
Db	Decibelios

ÍNDICE DE FIGURAS

Figura 1.1 Diagrama General	3
Figura 2.1 Diagrama del sistema de Monitoreo	14
Figura 2.2 Topología utilizada	16
Figura 2.3 presentación del IDE Arduino	17
Figura 2.4 Conexión con red Wi-Fi	18
Figura 2.5 Declaración de constantes FIREBASE_HOST y FIREBASE_AUTH: Dirección URL y TOKEN, inicio de comunicación	18
Figura 2.6 Escritura de datos en base de datos	19
Figura 2.7 Consola de creación y acceso de proyectos.....	21
Figura 2.8 Opciones de aplicaciones Firebase	21
Figura 2.9 Presentación de datos en la Base de Datos en Tiempo Real	22
Figura 2.10 Creación de variables y referencias.....	22
Figura 2.11 Estructura del EventListener.....	23
Figura 2.12 View Pager	24
Figura 2.13 ViewHolder	25
Figura 3.1 Comparación entre el modelo propuesto por [18], modelo Montley-Simplificado y valores experimentales.....	28
Figura 3.2 Cantidad de tiempo que demora en enviar la información del módulo a la base de datos en tiempo real Firebase	29
Figura 3.3 Probabilidad de errores producidos en un periodo de 5 horas.....	31
Figura 3.4 Interferencia co-canal producida por dispositivos Wi-Fi.....	34
Figura 3.5 Diagrama de barras de la pregunta # 1	36
Figura 3.6 Diagrama de barras de la pregunta # 2	37
Figura 3.7 Diagrama de barras de la pregunta # 3	38
Figura 3.8 Diagrama de barras de la pregunta # 4	39

ÍNDICE DE TABLAS

Tabla 4.1 Costos de Elaboración Dispositivos monitoreo	35
Tabla 4.2 Costos de Elaboración Dispositivos monitoreo para una sala.....	35

CAPÍTULO 1

1. Análisis de componentes tecnológicos y problemática.

En este trabajo se analizarán los problemas que existen en el país, relacionado con las enfermedades respiratorias tales como la neumonía, bronquitis, influenza y demás. Cabe destacar que el análisis se realizó para una población de infantes de 0 a 4 años que representan uno de los sectores más vulnerables del país.

Las estadísticas de la Organización Mundial de la Salud reflejan que las diez causas más frecuentes de muerte en las personas a nivel mundial en el año 2016 están [1]:

- Enfermedades cardíacas.
- Derrames.
- Enfermedades crónicas obstructivas pulmonares.
- Infecciones respiratorias menores.
- Enfermedad de Alzheimer y otras.
- Cáncer de pulmones, bronquios y tráqueas.
- Diabetes.
- Accidentes de tránsito.
- Enfermedades diarreicas.
- Tuberculosis.

Como se puede observar, varias de estas causas están relacionadas a problemas respiratorios, además, las infecciones respiratorias menores son producidas, o tienen como causas, el hecho de ser contagiadas por condiciones nutricionales, de manera maternal o neonatal [1]. Analizando en el Continente Americano tenemos que la influenza y neumonía son la primera causa de muerte en la subregión andina, cabe destacar que esta región la comprenden Ecuador, Bolivia, Venezuela, Colombia y Perú. En términos estadísticos representan el 12.5% de muertes estimadas en esta región [2].

En el caso particular de Ecuador, tenemos que la influenza y la neumonía ocupan el cuarto lugar de las causas de muertes de las diez más frecuentes, por debajo de las enfermedades de corazón, la diabetes y las enfermedades cerebrovasculares. Se estima que el 6.6% de las muertes en el país son causadas por infecciones respiratorias, normalmente estas infecciones son estacionarias, es decir, que se dan en una estación climática determinada [3]; podemos ver según las estadísticas presentadas en los párrafos anteriores, que a nivel global estas infecciones están siempre presentes como causantes del deceso de millones de personas alrededor del mundo, adicionalmente la región subandina en la que se encuentra el país es un sector geográficamente vulnerable a estas enfermedades y, más aún, cuando las condiciones climáticas ayudan a la proliferación de estas.

En el caso concreto del país tenemos que así mismo tienen una alta tasa de mortalidad entre los ciudadanos y como se mencionó, uno de los grupos más vulnerable son los infantes, si a este problema sumamos la falta de personal médico en los hospitales (el cual será sustentado de manera adecuada más adelante) podemos ver que esto es propicio para las elevadas cifras de mortalidad por estas enfermedades.

Es por lo que se propone una solución para este problema desde el punto de vista del monitoreo de los pacientes y se analiza el impacto que tiene la solución propuesta como un soporte o ayuda a los médicos que están encargados de los infantes con síntomas de estas enfermedades, así mismo el diseño de un prototipo que va a ser capaz de ayudar con la tarea encomendada. Puesto que el número de médicos es muy inferior a la cantidad de habitantes del país, se puede estimar la cantidad de doctores residentes que están atendiendo a los infantes internados. El sistema que se propone no es más que una forma de optimizar la cantidad de personas encargadas del cuidado de los pacientes, para así mejorar el sistema de respuesta ante cualquier eventualidad que se pueda dar.

En este capítulo presenta y sustenta, en base a datos estadísticos, el problema que representan las enfermedades respiratorias en los infantes, de la misma manera la cantidad de médicos por pacientes y se propone una solución para el problema que representa la baja cantidad de médicos por pacientes. De manera adicional, se

enuncian los objetivos a cumplir en este trabajo y se presenta la teoría correspondiente a las tecnologías y los componentes a utilizar.

Para entender mejor la visión de nuestro proyecto, procedemos a mostrar un diagrama simplificado.

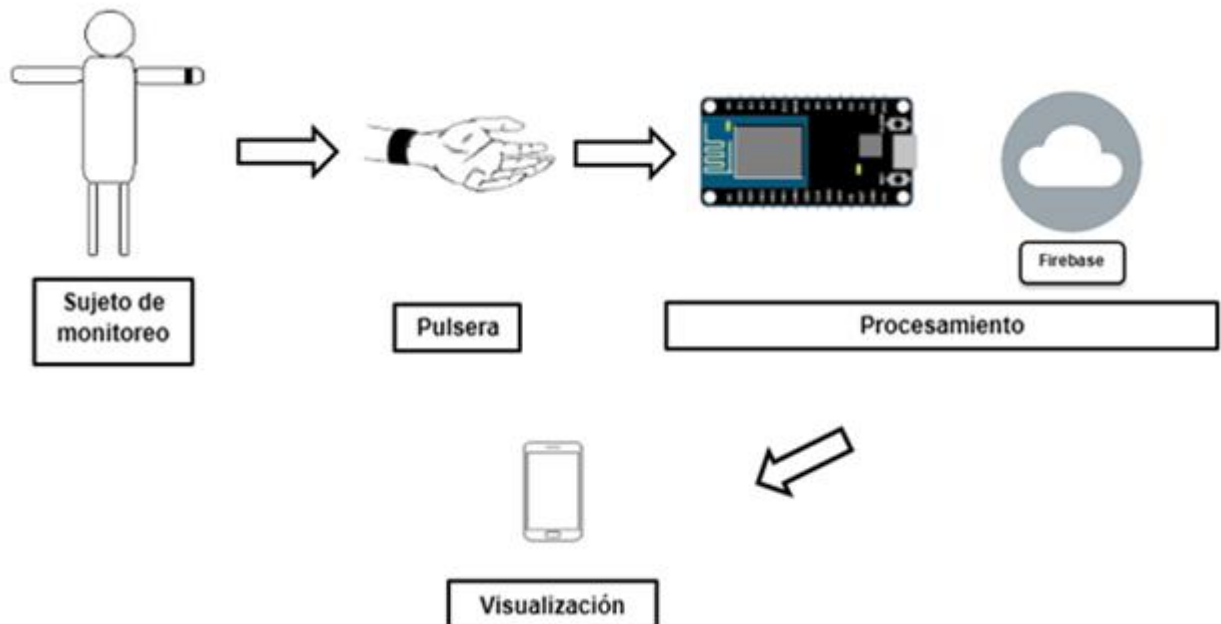


Figura 1.1 Diagrama General

La figura 1.1 muestra los bloques que representan de manera simplificada cada una de las etapas que conforman el presente proyecto, las cuales se procederá a explicar a continuación:

Sujeto de monitoreo: Pacientes que se encuentran de hospitalizados, los cuales presentan enfermedades respiratorias, cuyos síntomas nos dieron la pauta de sensor ciertos signos vitales como lo es la temperatura y el porcentaje de oxígeno en la sangre. Cabe aclarar que este diagrama es la representación del monitoreo de un paciente, pero el mismo principio es utilizado para un mayor número pacientes.

Pulsera: Dispositivo cuya función es mantener los sensores de temperatura y porcentaje de oxígeno estables y en una ubicación específica para la correcta toma de los signos vitales. Su tamaño y peso debe ser lo más pequeño posible, se escogió el modelo pulsera para nuestro proyecto debido a que es más sencillo colocarla, permite variar el tamaño para adaptarse a cualquier tamaño de muñeca y genera poca incomodidad a los pacientes.

Procesamiento: Este proceso involucra tanto el desarrollo como el tratamiento de la información, así como las plataformas y programas que serán necesarios para cumplir con los requerimientos.

Visualización: Puede ser realizada por medio de una computadora o dispositivo móvil para mayor comodidad.

El proceso detallado es mostrado y explicado en el capítulo 2.

1.1. Planteamiento del problema

Según cifras de la Organización Mundial de la Salud (OMS), una de las causas de muerte más elevadas en infantes de edades entre 0 y 4 años son las infecciones respiratorias, dejando de lado casos como partos prematuros, traumas o eventualidades en el parto y anomalías congénitas tenemos que las enfermedades respiratorias presentan el mayor índice de mortalidad en los infantes [4].

La población del Ecuador hasta el año 2016 fue de 16'528.730 [5], de los cuales los infantes entre 0 a 4 años representan cerca del 9%, siendo una población de 1'469.042. Si tomamos como referencia las estadísticas presentadas por la OMS, tenemos que mueren 5,1 infantes por cada 1000; esto quiere decir que de la población total de infantes de 0 a 4 años en el 2016 han fallecido 7.492,11. Según estadísticas del INEC del año 2014 fueron 81.414 personas que egresaron de los hospitales debido a afecciones respiratorias, de este total 46.360 se dividen afecciones respiratorias agudas, neumonía de diferentes tipos, influenza y bronquitis. La población de infantes de 0 a 4 años egresados fue de 31.763, en este caso 26.180 infantes que egresaron fueron recibidos por las enfermedades mencionadas en el párrafo anterior [6].

1.2. Objetivos

1.2.1. Objetivo General

- Diseñar un sistema que permite enlazar diferentes equipos telemédicos, utilizando tecnologías IoT (por sus siglas en inglés internet de las cosas), para el monitoreo en tiempo real de pacientes con enfermedades respiratorias por medio de una conexión a internet.

1.2.2. Objetivos Específicos

- Obtener la información de cada uno de los sensores de signos vitales (temperatura y ritmo cardiaco), para la manipulación de los datos en el HiLetgo encargado del procesamiento.
- Enviar la información a una plataforma IoT para su posterior acceso en cualquier lugar con una conexión a internet, por medio de un tablero de control la visualización y a su vez la recepción de las alertas emitidas por los dispositivos.
- Diseño de un prototipo telemédico, en el cual se integran los sensores y el HiLetgo, para el procesamiento de los signos vitales en pacientes de 0 a 4 años.

1.3. Justificación del problema.

En las estadísticas obtenidas por el INEC del año 2014 podemos evidenciar que del total de egresos el 82,42% fueron tratados por enfermedades respiratorias, adicional a esto tenemos que del total de la población existían en ese año 32.617 médicos. Comparando la población del país en esa fecha se estimó que por cada 10.000 habitantes había 20 médicos [6].

Si comparamos el dato del número de médicos, con los datos de ingresados del mismo año en los cuales se atendieron a pacientes con enfermedades respiratorias, tenemos que los 46.360 pacientes fueron posiblemente atendidos por 94 médicos, evidenciando la falta de personal para poder tener un control óptimo de los pacientes con problemas respiratorios [6].

Debido a la carencia de personal en los hospitales y el número de muertes de niños por enfermedades respiratorias, hemos procedido a la búsqueda de una solución, cuyo objetivo es ofrecer al doctor una herramienta que permita monitorear a sus pacientes en cualquier lugar por medio de una aplicación móvil, además de contar con una alarma que le permita saber cuál signo vital esta fuera del rango normal, enviando un mensaje al tablero de control.

Se procede a obtener los signos vitales por medio de sensores los cuales están conectados a la tarjeta de adquisición, luego estos datos son procesados y enviados a un servidor para posteriormente ser mostrados en una aplicación.

El presente proyecto se encuentra distribuido de la siguiente manera: en el Capítulo 1 se muestran las bases y principios en lo cual nos basamos para realizar el proyecto. El Capítulo 2 se centrará en la forma en el que atacamos el problema y cada uno de los pasos necesarios para llevar a cabo el proyecto. Finalmente, en el Capítulo 3 explicamos las ventajas de esta tecnología utilizada y por medio de cálculos mostramos la factibilidad y utilidad de nuestro proyecto.

1.4. Marco Teórico

1.4.1. Sensores

Los sensores son dispositivos diseñados para captar estímulos externos y responder en consecuencia, debido a que tienen esta propiedad se pueden utilizar para diferentes tipos de aplicaciones ya que la función de ellos es convertir una magnitud física (recibida por el captador) en una magnitud eléctrica (este proceso se efectúa gracias al transductor).

Sensor de temperatura. Cabe indicar que además del termómetro de mercurio existen diferentes formas de conocer la temperatura, por ejemplo; el termistor NTC es una resistencia que varía su valor óhmico dependiendo de la temperatura a la cual se exponga, el termostato bimetalico son dos placas unidas con coeficiente de dilatación térmica diferente, en el caso del termopar se sueldan los extremos de metales diferentes cuya unión actúa como sonda para conocer su temperatura. Para fines de nuestro proyecto utilizaremos el termómetro digital sin contacto.

Termómetro infrarrojo digital sin contacto (MLX90614). Este módulo es un termómetro infrarrojo que se utiliza para mediciones de temperatura sin contacto, tanto el acondicionador de la señal como el chip detector de termopila se encuentran en la misma placa. Cuenta con una salida SMBus que da acceso a la temperatura con una resolución de 0.02°C. Se puede configurar la salida digital para que sea modulación por ancho de pulso

(sus siglas en inglés PWM) [15]. Según la ley de Stefan-Boltzman, todo objeto cuya temperatura se encuentre sobre el cero absoluto emite radiación proporcional a la temperatura de esos objetos en su campo de visión. El MLX90614 consta de un chip de silicio con una fina membrana la cual es sensible a la radiación infrarroja, junto con la electrónica necesaria para amplificar, digitalizar la señal y calcular la temperatura.

Oxímetro de pulso (MAX30100). Es un dispositivo que permite medir de manera indirecta las pulsaciones, detectando la cantidad de oxígeno en la sangre. Al pasar la sangre oxigenada por los pulmones, la hemoglobina (Hb) se convierte en oxihemoglobina (HbO₂), de esta manera se hace posible transportar el oxígeno por la sangre. Estos compuestos tienen diferentes niveles de absorción para diferentes longitudes de onda de la luz. Entre los 650nm (rojo) y 950nm (infrarrojo) la diferencia en el comportamiento de la oxihemoglobina y la hemoglobina se hace más fácil distinguir, hasta llegar a 800nm, la hemoglobina absorbe más luz (roja) y a partir de este punto se invierte, de esta manera la oxihemoglobina absorbe más luz (infrarroja).

Para medir la existencia de sangre oxigenada se emite una luz roja y se detecta la intensidad que se atraviesa y luego se procede a hacer lo mismo con la luz infrarroja, en función de las intensidades absorbidas se puede conocer el nivel de oxígeno [16].

1.4.2. Tarjeta controladora Hiletgo.

Para nuestro proyecto utilizaremos kit de código abierto que tiene integrado Wi-Fi. El uso del Wi-Fi nos permite conectarnos mediante un enlace inalámbrico a una router con conexión a internet. Una gran ventaja de este módulo es que tiene potentes capacidades de almacenamiento y procesamiento, que le permiten interactuar con sensores y dispositivos a través de sus entradas/salidas de propósito general (sus siglas en inglés GPIOs).

Su alto grado de integración nos permitirá una circuitería externa mínima, la corriente de reposo es inferior a 5uA adecuada para nuestra aplicación, debido a que utilizaremos baterías. Consta de 2 núcleos los cuales pueden ser controlados individualmente, se puede apagar el CPU y utilizar el coprocesador de baja potencia para la supervisión constante de algún cambio de estado. Como especificación se debe tener en consideración que trabaja a 3.3V. El uso de esta tecnología permite la posibilidad de realizar todo tipo de prototipos debido a su bajo costo y a la gran cantidad de sensores disponibles en el mercado. La utilización de esta tecnología se analiza en la Sección 2.2.

1.4.3. Tecnología Wi-Fi

Wi-Fi es una abreviatura de las palabras Wireless Fidelity, esta marca comercial es la encargada de certificar equipos que cumplen con varios estándares tales como: IEEE 802.11b, IEEE 802.11g e IEEE 802.11n. Cuando un equipo adopta estas tecnologías, la empresa pasa a ceder su logo como indicativo que cumple a cabalidad con las condiciones que exigen los estándares de utilización.

El IEEE (Institute of Electrical and Electronics Engineers) tiene varias recomendaciones que se adoptan a las tecnologías inalámbricas locales según las necesidades que se presenten. Algunas de estas son:

- IEEE 802.11b.
- IEEE 802.11g.
- IEEE 802.11n.
- IEEE 802.11ac.

La arquitectura del protocolo 802.11 se basa en el set de servicios básico (BSS por sus siglas en inglés) que es el bloque fundamental en la construcción de este. El BSS es definido como un grupo de estaciones que está bajo control directo de una función simple de coordinación [7].

El área geográfica de cobertura por una BSS es conocida como área de servicio básico (BSA por sus siglas en inglés), se puede hacer una analogía de estos elementos a la construcción de una red móvil, la cual

está constituida por celdas. Los conjuntos de servicios básicos pueden comunicarse entre sí de manera directa, sin embargo, al tener en consideración todos los elementos causantes de la degradación de las señales tales como, el desvanecimiento por multitrayecto en los medios de transmisión, las pérdidas por el tipo de ambiente, etc. es probable que algunos de estos puedan estar ocultos para la perspectiva de otras estaciones BSS [7].

Existen diferentes formas de intercomunicación entre las estaciones, una de ellas es la red ad hoc. La red ad hoc consiste en un grupo de estaciones dentro de una BSS sin la necesidad de tener una infraestructura de red. Esta forma de comunicación entre redes es conocida como BSS independiente (IBSS por sus siglas en inglés) por el estándar establecido en el IEEE 802.11. Otra forma de comunicación entre estaciones en un BSS y sin requerimientos para canalizar todo el tráfico, es a través de un punto de acceso centralizado (AP por sus siglas en inglés). Si bien es cierto este estándar es para redes locales inalámbricas, se puede hacer una analogía con las redes celulares, en la cual la estación base bien podría ser una representación de los APs, esto quiere decir que los APs son los responsables de proveer los puntos de integración necesarios para la conectividad de red entre varias BSS, formando así los conjuntos de servicios extendidos (ESS por sus siglas en inglés) [7].

Como se mencionó, la integración de varios BSS permite la creación de los ESS; pero esto es posible solamente a que el sistema de distribución (DS por sus siglas en inglés) es usado en la interacción de las BSS. Este DS es el responsable de del transporte a nivel de la MAC de las unidades de datos de servicios de MAC (MSDUs por sus siglas en inglés), además de lo estipulado en el documento del estándar IEEE 802.11 este sistema es independiente de la instalación. Sin embargo, la forma en la que está constituido este sistema le permite interactuar con varios protocolos de comunicación, tales como, IEEE 802.3 que hace referencia a redes Ethernet LAN, IEEE 802.4 el cual está implementado sobre cable coaxial y tiene la curiosidad de ser una red lógica con topología de tipo bus, IEEE

802.5 el cual tiene similitudes con el protocolo anterior solo que la topología de este es del tipo anillo, interfaz de datos distribuidos por fibra (FDDI por sus siglas en inglés), redes de áreas metropolitanas o cualquier otro medio que use el protocolo IEEE 802.11. En términos físicos el DS podría estar en el mismo medio de transmisión que un BSS, pero si hablamos en términos lógicos difieren en el hecho de que el DS es meramente una columna de transporte para transferir paquetes de diferentes BSS en los ESS [7].

En la referencia [7] existen tres diferentes niveles de implementación en la capa física, dos de estas implementaciones trabajan en la banda de los 2.4 GHz y la otra trabaja en una longitud de onda de 850 a 950 nm. La primera de estas corresponde al espectro de salto ensanchado de frecuencia (FHSS por sus siglas en inglés), este método tiene como rango de trabajo al definido por IEEE para las áreas científicas, industriales y médicas. Además de que posee un espaciamiento entre canales de 1 MHz, el FHSS tiene una tasa de acceso básica de 1 Mbit/s, para que se pueda dar esta tasa de bits se utiliza una modulación llamada “desplazamiento gaussiano de frecuencia” o (GFSK por sus siglas en inglés). Esta modulación lo que hace es que se codifican bits en un determinado intervalo de frecuencias [7].

La segunda corresponde al espectro ensanchado por secuencia directa (DSSS por sus siglas en inglés), que trabaja en la banda de los 2.4 GHz ISM pero su diferencia está en el tipo de modulación utilizado. La modulación por desplazamiento de fase binaria diferencial (DBPSK por sus siglas en inglés) es la que se utiliza para poder conseguir una tasa de datos de 1 Mb/s. Si se quiere aumentar la tasa de datos se debe elevar el número de constelaciones en la modulación, o niveles tomando en cuenta la GFSK. El desplazamiento se logra al dividir todo el ancho de banda disponible en once subcanales con un ancho de 11 MHz que, con la ayuda de la secuencia 11 chip - Barker para poder distribuir cada símbolo en cada uno de los canales, se puede evitar que se produzcan traslapes y adyacencias

entre las BSS de una ESS asegurándose que cada BSS tenga espaciamientos entre las frecuencias centrales al menos 30 MHz [7].

La tercera y última implementación a nivel físico que propone el protocolo IEEE 802.11 es la IR que como se mencionó antes, trabaja en el rango de los 850-950 nm. Esta solución física planteada es aconsejable usarla solo para interiores y en transmisiones no directas. Según las especificaciones es necesario que exista línea de vista y transmisión reflejada; toda la tasa de 1Mb/s se basa en una modulación de posición de pulsos (PPM por sus siglas en inglés) donde 4 bits de datos son mapeados a 16 bits codificados para su posterior transmisión [7].

1.4.4. Plataforma IoT

Internet de las cosas o Internet of things (por sus siglas en inglés IoT), es una forma revolucionaria de aprovechar las facilidades que ofrece hoy en día la tecnología con respecto a la transmisión de información. El uso de dispositivos móviles u objetos de la vida cotidiana que se conectan e interactúan entre sí por medio de comunicaciones inalámbricas, ha permitido que se desarrollen diferentes campos en base a esta implementación que está en continuo auge; parte importante del desarrollo de las aplicaciones de IoT radica en la constante actualización de la infraestructura de redes inalámbricas, por ejemplo la migración de IPv4 a IPv6 por motivos de espacio en la asignación de direcciones en la versión 4 de esta red.

Las aplicaciones más comunes para IoT son en la parte de monitoreos o controles ambientales para edificios y casas por medio de sensores y/o cámaras; en algunos trabajos como [8] podemos ver que estos sistemas buscan una forma de converger por las facilidades que IPv6 le brinda desde el punto de vista de acceso. En el mismo documento podemos apreciar que se ha trabajado en el concepto de tener edificios automatizados y conectados a internet, normalmente lo que se busca con esto es poder aprovechar mejor los recursos evitando incrementar los costos o sacrificar la eficiencia. En el trabajo [9] se habla de cómo los

sensores de red inalámbrica pueden ser implementados para monitorear las condiciones normales del hogar; en este trabajo podemos ver como se hace un uso efectivo de las tecnologías inalámbricas disponibles tales como el ZigBee. Por medio del gateway se hace una traducción del protocolo de formato de datos del ZigBee al protocolo IP en su versión 6.

Como se ha podido observar, IPv6 e IoT están relacionados según los trabajos expuestos en párrafos anteriores, estos hacen referencia al uso genérico de IoT. El punto de interés en esta innovadora forma de aplicación radica en las aplicaciones telemáticas. En [10] se muestra una aplicación para el control y acceso de la información presentada en un hospital, como puede ayudar a mejorar los problemas en el registro manual de los datos o fijar un punto de información. Para estos problemas expuestos en el trabajo mencionado se han presentado diferentes soluciones como, por ejemplo, identificación por radiofrecuencia (RFID por sus siglas en inglés), sensores infrarrojos, GPS, escáneres láser y otra forma de obtener la información. Como tecnologías claves en este trabajo están: IoT, RFID, tecnología de red de sensores, comunicación inalámbrica y sistemas embebidos. En el trabajo [11] también se puede ver que se apunta a integrar las tecnologías mencionadas anteriormente como lo son IoT, RFID y demás, en este trabajo se explica de qué manera se pueden integrar los instrumentos médicos a una red de internet, además de los diferentes tipos de aplicaciones a los diferentes campos que se presentan en la medicina. Siguiendo las áreas expuestas en [11] tenemos que dividen la telemedicina en consulta interactiva y consulta no interactiva. En la primera se plantea la idea de tener una consulta de manera remota con un experto en el área de interés por medio de videoconferencias para hacerlo de una manera “presencial”. En la consulta no interactiva el experto lee, analiza y estudia los datos enviados por un paciente a través de una red.

Las ventajas que se tienen del uso de IoT en la medicina ayudan a mejorar drásticamente ciertas condiciones tales como: i) el costo de traslado de pacientes a los centros de salud, ii) manejo y distribución de recursos médicos en zonas de difícil acceso, iii) la diversificación en la información

y las consultas pueden realizarse desde cualquier lugar, y iv) se elimina el problema de las barreras geográficas.

En [11] el autor comenta los problemas que se presentan en China respecto al uso de las redes ya existentes, siendo un país industrializado la calidad de transmisión no es la mejor, el ancho de banda es parecido a un cuello de botella, los protocolos y estándares de comunicación entre los hospitales no permiten que interactúen entre las diferentes instituciones, el costo de la implementación de los sistemas no permite que sea posible un sistema universal de consultas y el tamaño de algunos datos e imágenes médicas puede ser demasiado grande para el procesamiento de estos datos.

La implementación de RFID ayuda para el etiquetado de las muestras, alarmas en caso de anomalías o administración correcta de los equipos disponibles, así es posible tener un mayor control tanto en pacientes como en los implementos médicos usados.

Capítulo 2

2. Sistema de sensado y manejo de la información

En este capítulo se procederá a la descripción de los elementos que forman parte de la red propuesta, se muestra cada uno de los componentes del sistema y cómo interactúan cada uno de ellos. En la propuesta que se realiza en este trabajo se tiene una red que funciona de la siguiente manera: un equipo que sirve para la captura de los datos de interés para el médico con respecto a los síntomas de enfermedades respiratorias, el envío de datos a través de este dispositivo electrónico a una plataforma web donde se realiza el monitoreo en tiempo real, y el desarrollo de una aplicación celular, la cual es la responsable de mostrar los datos en tiempo real al médico para el control de los pacientes con el dispositivo electrónico. Así como se muestra en la figura 2.1

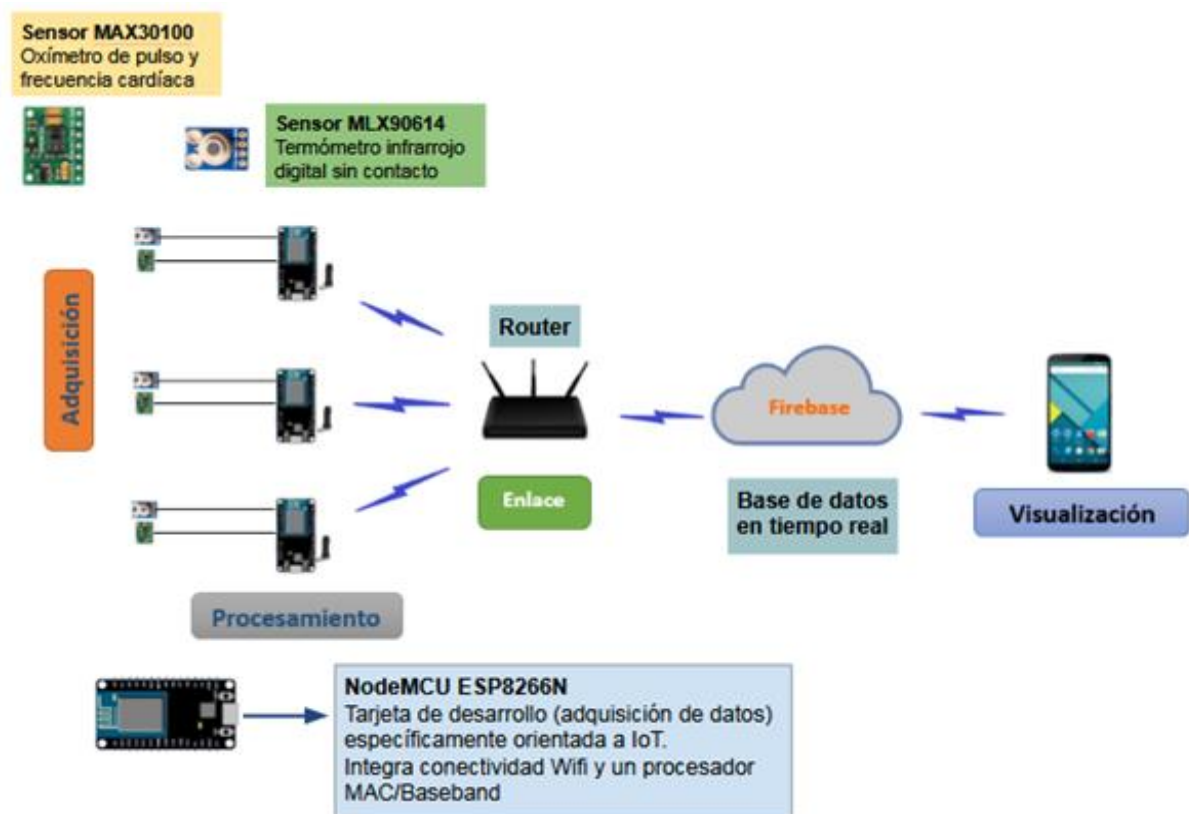


Figura 2.1 Diagrama del sistema de Monitoreo

La figura 2.1 muestra la etapa de adquisición en la cual tenemos los sensores que vamos a utilizar para adquirir los signos vitales (temperatura y porcentaje de oxígeno en la sangre), estos sensores se encuentran conectados al módulo (HiLetgo).

Luego de adquirir los datos pasamos a la etapa de procesamiento cual se encarga el módulo HiLetgo mismo que utilizando programación, ordena los datos y lo enviamos a la plataforma (Firebase) a través del módulo Wi-Fi interno. La comunicación entre el módulo y la plataforma se realiza de manera inalámbricas (Wi-Fi) por medio del enlace (módulo - router). La plataforma (Firebase) se utiliza para comunicar los datos en tiempo real entre el módulo y la aplicación. Para la visualización se utiliza un dispositivo móvil, dado que es un medio muy utilizado y permite que cualquier usuario puede visualizar la información con solo instalar la aplicación.

2.1. Topología de la red.

Nuestro proyecto se basa en el monitoreo de una sala de hospital trabajaremos con la primicia que existe más de un paciente en la zona de estudio, por esta razón hemos visto necesario la utilización la topología “estrella” como podemos observar en la figura 2.2 para la comunicación entre los módulos y la plataforma. Esta configuración brinda muchas ventajas al momento de existir alguna falla en los sensores o en módulo “HiLetgo”. Al producirse alguna falla en uno o varios módulos, ya sea por estar desconectado o dañado, sigue sensando a los demás pacientes sin dar inconvenientes a los otros dispositivos dado que en esta topología cada módulo es conectado directamente a la plataforma IoT.

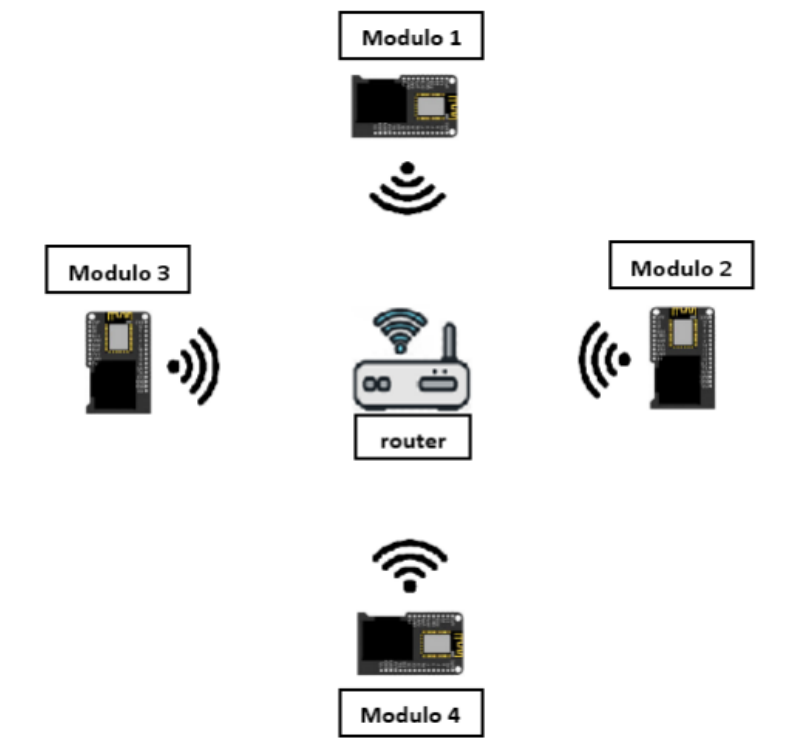


Figura 2.2 Topología utilizada “estrella”

2.2. Manejo de la información

Los dispositivos electrónicos que intervienen en prototipo son los responsables de la adquisición de los parámetros vitales, que resultan ser variables de interés de los médicos, así como los que transportan o envían esos datos a la siguiente etapa de la red.

2.3.1. NodeMCU

Este dispositivo electrónico es un microcontrolador que, además, posee un módulo Wi-Fi incorporado lo que hace idóneo el procesamiento y transmisión de datos en la banda de 2.4GHz. La tarjeta cuenta con entradas y salidas analógica las cuales interactúan con los sensores responsables de tomar las muestras según el tipo de datos que estos arrojen.

Para el correcto funcionamiento de cada una de las partes el programa que se utiliza para su programación fue el IDE ARDUINO, este programa posee una estructura la podemos ver en la figura 2.3.

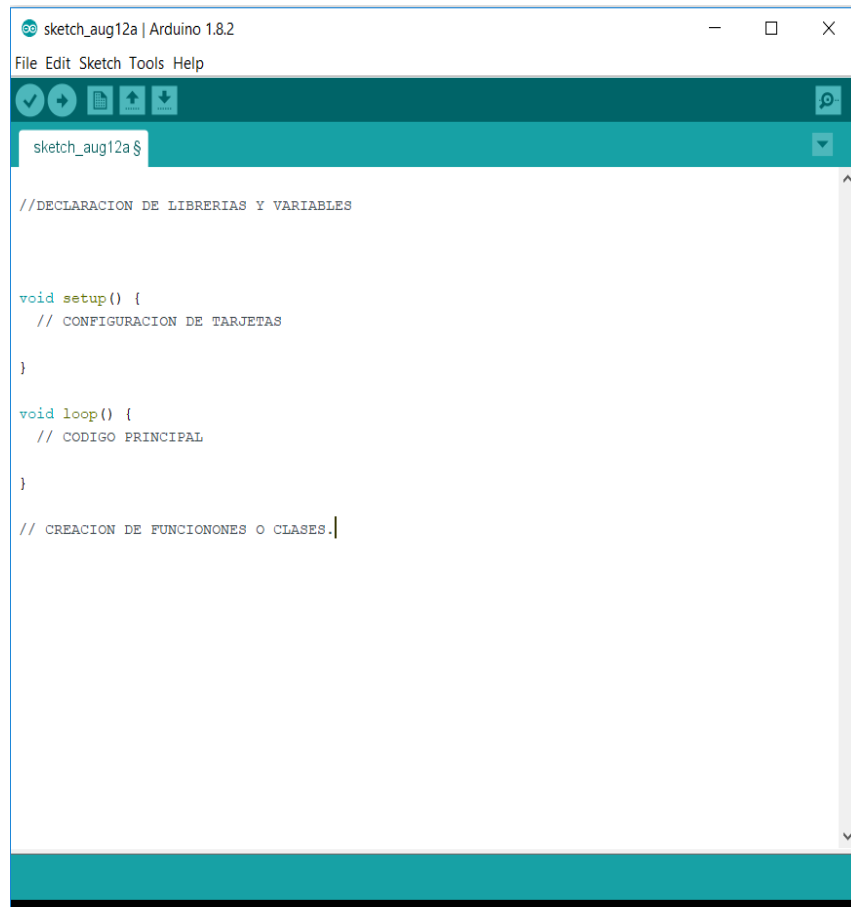


Figura 2.3 presentación del IDE Arduino

En este programa siempre es necesario la declaración de las librerías puesto que estas son las que tienen las funciones necesarias a utilizarse según las necesidades del programador. En este caso se integraron dos librerías las cuales son: `#include <ESP8266WiFi.h>` y `#include <FirebaseArduino.h>`.

La primera librería sirve para establecer la comunicación entre el módulo y la red de internet la cual está disponible en el sitio donde se utilizan los equipos electrónicos. Cabe destacar que el responsable de establecer la comunicación inalámbrica es el módulo Wi-Fi ESP8266, este forma parte de la tarjeta nodeMCU. En esta librería se procede a establecer la configuración de la red Wi-Fi, así como la conexión al router disponible para la comunicación.

La librería `FirebaseArduino.h` es la responsable de poder establecer una vía de transmisión entre la base de datos y el dispositivo en tiempo real. Si bien es cierto la librería `FirebaseArduino` se encuentra en una etapa de desarrollo, funciona bastante bien para las aplicaciones que se necesitan en esta red que se está proponiendo.

En la configuración se recibe como parámetros el nombre de la red y la clave si es que existe, de manera adicional una vez que los datos han sido autenticados se establece la comunicación y así el dispositivo electrónico puede comunicarse con la base de datos en tiempo real y tener salida a la red.

```
void setup() {
  Serial.begin(9600);

  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
}
```

Figura 2.4 Conexión con red Wi-Fi

En cuanto a la programación primero se necesita indicar a donde nosotros necesitamos apuntar para poder tener el acceso de lectura y escritura de datos, posterior a esto es necesario verificar si existe o no la interacción entre estos dos elementos, tanto en el componente electrónico como en la plataforma encargada de la recepción y transmisión de datos en tiempo real.

Para poder tener el acceso a la base de datos se dispone de un enlace y de un token o clave secreta, de esta forma ingresando estos datos se puede saber a donde nosotros queremos que se registren los datos.

```
#define FIREBASE_HOST "prueba-8415d.firebaseio.com"
#define FIREBASE_AUTH "kMxR8UsrFrLrBvTFRkeGX5cbhW37h"
```

Figura 2.5 Declaración de constantes `FIREBASE_HOST` y `FIREBASE_AUTH`: Dirección URL y TOKEN, inicio de comunicación

En la figura 2.5 podemos observar cómo se definen las constantes que establecen la dirección web a donde se van a registrar los datos que serán enviados. FIREBASE_HOST contiene la dirección de la base de datos, la constante FIREBASE_AUTH es la clave que concede los permisos de acceso a la base de datos en tiempo real.

Después de establecer la conexión entre el dispositivo y la base de datos se procede a iniciar la comunicación entre la base de datos y la nodeMCU, que como se explicó con anterioridad es la encargada de recibir los parámetros y enviarlos a la base de datos en tiempo real. Después de que la comunicación es establecida se configura la manera en que ambos elementos interactúan entre ellos, dado que estos son de vital importancia en esta red propuesta.

Una vez que se ha establecido correctamente la conexión a la red y a la base de datos por parte de la tarjeta, se procede a inicializar el código principal de la tarjeta electrónica, es aquí donde se procede a la escritura y al envío de datos a la base de datos para que puedan ser leídos.

```
    Firebase.setInt("Paciente 1/SO2",n++);
    Firebase.setInt("Paciente 1/P.A SIST",n++);
    Firebase.setInt("Paciente 1/P.A DIAST",n++);

    // handle error
    if (Firebase.failed()) {
        Serial.print("pushing /logs failed:");
        Serial.println(Firebase.error());
        return;
    }
```

Figura 2.6 Escritura de datos en base de datos

La figura 2.6 muestra los datos que van a ser utilizados en la plataforma Firebase, como podemos ver en la figura 2.6 en las funciones *setInt* de la librería *FirebaseArduino* tiene como parámetros de entrada: el camino al cual se van a registrar los datos, y el valor numérico a registrar en la base de datos. La estructura de estos datos en la base son de la forma Notación de Objetos de JavaScript (por sus siglas en inglés JSON), esto quiere decir

que de manera pura se pueden representar como un árbol cuya ramificación se puede tener los hijos de la raíz o dato primario. En esta forma de presentar los datos se puede direccionar la lectura o escritura de manera directa, evitando así que existan problemas en que los datos se sobrepongan entre sí.

La estructura condicional en el código, muestra un error al momento en que el dato no puede ser enviado desde la tarjeta electrónica a la base de datos, esta es la manera en la que se puede detectar errores en la transmisión de los datos y así evitar enviar código basura a la base de datos. La diferencia entre hacer un *setInt* con otra de las funciones que se tiene como lo es *pushInt*, es que al momento de hacer *push* los datos ya van con una etiqueta en el hijo de la raíz determinada; es decir que no se puede personalizar la presentación de los datos a la persona que se le van a mostrar los valores de interés.

La función *setInt* da la oportunidad de poder customizar las etiquetas, poder darle los identificadores adecuados a la base de datos y así poder hacer más presentable los datos a las personas responsables del manejo de los valores que serán presentados.

2.3.2. Google Firebase (Base de datos en tiempo real)

Esta plataforma de Google tiene algunas opciones interesantes para el desarrollo de aplicaciones IoT, siendo la de “RealTime DataBase” la opción escogida para el desarrollo de nuestra base de datos en tiempo real. Esta opción permite registrar y leer datos que sean enviados o ingresados en el proyecto que se va a crear.

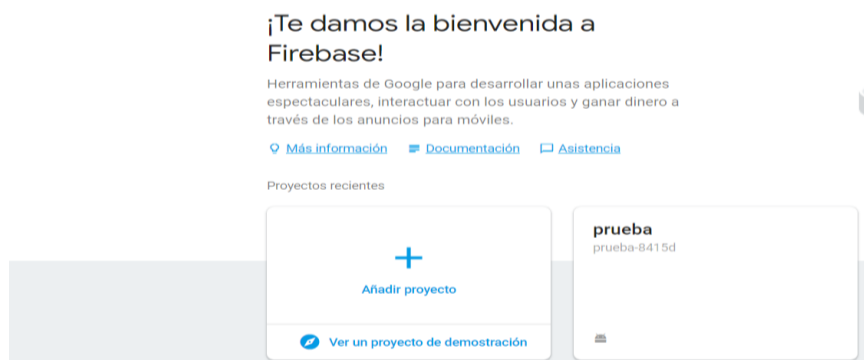


Figura 2.7 Consola de creación y acceso de proyectos

La figura 2.7 muestra la página de inicio de la plataforma Firebase, aquí es donde se crea un proyecto y se lo procede a registrar para el desarrollo de cualquiera de las opciones que se integra en esta plataforma. Además, después de que se crea el proyecto se definen las reglas que van a regir la comunicación con cualquier aplicación externa a la plataforma Google Firebase.

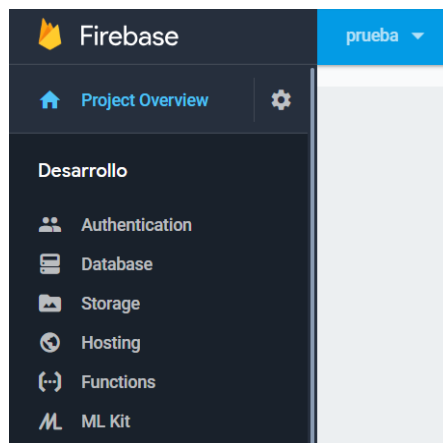


Figura 2.8 Opciones de aplicaciones Firebase

Dentro de las opciones mostradas en la figura 2.8, la de base de datos es la que se ajusta a las necesidades que presenta la solución propuesta, dado que se puede elegir la opción de que funcione en tiempo real, lo que es necesario para tener los cambios al mismo instante que la nodeMCU los detecta. La importancia de la base de datos en tiempo real radica en casos donde se manejan pacientes vulnerables cuyo seguimiento debe ser constante, donde cualquier cambio en la condición del paciente es significativamente importante. El hecho que una variable de interés cambie y se produzca algún retraso significativo en la presentación de estos datos puede ser motivo de alguna emergencia o situación no deseada, la cual puede conllevar problemas crónicos.

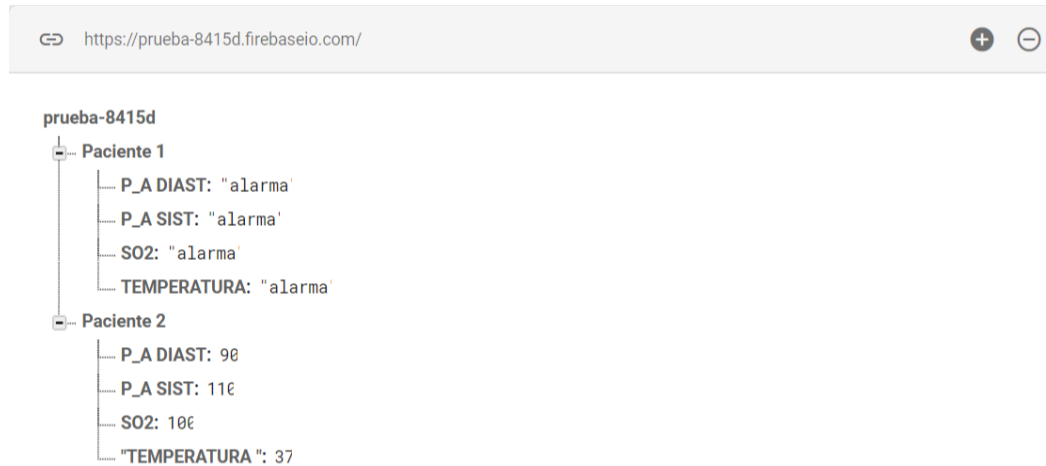


Figura 2.9 Presentación de datos en la Base de Datos en Tiempo Real

2.3.3. Visualización (Aplicación móvil)

La aplicación móvil es desarrollada con el programa “Android Studio”, en este programa se desarrolla la lectura de los datos provenientes de la base de datos, para que el usuario tenga acceso a estos desde cualquier parte donde pueda tener conexión a internet; además de mostrar los datos de una manera que puedan ser fácilmente observados por los médicos que llevan a cabo el monitoreo de los signos vitales corresponden a cada paciente.

```

class MainActivity : AppCompatActivity() {
    class Paciente(var paciente: String, var sintomas: String) {...}

    var images: Array<Int> = arrayOf(R.drawable.descarga, R.drawable.imagen1)
    var listas: ArrayList<Paciente> = ArrayList()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val refer = FirebaseDatabase.getInstance()
        val abc = refer.getReference()
        var adapter = adapter(images, applicationContext, listas)
        viewPager.adapter = adapter
        abc.addValueEventListener(object : ValueEventListener {
  
```

Figura 2.10 Creación de variables y referencias

En la figura 2.10 se puede apreciar la etapa de declaración de variables, referencias y contenidos. Es importante vincular la aplicación móvil con la base de datos en tiempo real (Firebase) para poder acceder a los datos,

luego es necesario referenciar la base de datos, para que la aplicación pueda saber no solo a donde debe leer o escribir los datos, sino también a qué parte de todos los datos acceder para la posterior lectura.

Lo que es de interés en esta solución es la lectura de los datos, la función *addValueEventListener* es bastante importante puesto que es aquella que permite “escuchar” lo que se está produciendo en la base de datos en tiempo real. El uso de esta función abre dos posibilidades dentro del código: cuando se cancela un dato y cuando se produce un cambio en un dato.

```
    override fun onCancelled(p0: DatabaseError) {  
    }  
  
    override fun onDataChange(snapshot: DataSnapshot) {...}  
}}
```

Figura 2.11 Estructura del ValueEventListener

La estructura que merece la atención se produce en la función *onDataChange*, es aquí donde la aplicación registra los cambios que se produce en Firebase y los agrega a la parte de visualización en la aplicación móvil. Cabe mencionar que es necesario manipular de manera correcta los datos ya que en la parte del ValueEventListener se los recibe como un Snapshot y como se los desea presentar es en forma de cadena o String.

Para la parte de la estética en la aplicación se emplean dos métodos, estos son:

- ViewPager.
- ViewHolder.

El primero consiste en la creación de varios diseños para los fondos en la aplicación elaborada, cada página está destinada a los usuarios uno por uno, es decir, que a un usuario le corresponde una página con sus respectivos datos. Esto ayuda para obtener los datos de una manera

sencilla y clara para los responsables de los pacientes, además de que permite ciertas facilidades en la personalización de estas.

El segundo consiste en el hecho de poder mantener valores y escribirlos en listas. Esto ayuda a entregarlos de una manera ordenada y evitar que se sobrescriban los datos de los pacientes.

```
package com.tesis.rudyarb.prueba2

import ...

class adapter:PagerAdapter {
    var context: Context
    var images: Array<Int>
    var paciente: ArrayList<MainActivity.Paciente>

    lateinit var inflater: LayoutInflater

    constructor(images: Array<Int>, context: Context, paciente: ArrayList<MainActivity.Paciente>) : super() {...}

    override fun isViewFromObject(view: View, `object`: Any): Boolean = view == `object` as RelativeLayout

    override fun getCount(): Int = paciente.size

    override fun getItemPosition(`object`: Any): Int = images.size

    override fun instantiateItem(container: ViewGroup, position: Int): Any {...}
}
```

Figura 2.12 View Pager

Como se puede apreciar en la figura 2.12, las vistas de páginas se construyen en base a un adaptador, el mismo que recibe algunos parámetros de interés para poder manejarlos al momento de la presentación. En este caso las variables de entrada que reciben son arreglos de datos, los cuales están divididos en: imágenes, datos y el contexto que se refiere a el fondo de pantalla de la aplicación el cual van a ser presentados.

Este adaptador tiene funciones incorporadas, las cuales sirven para diferentes casos según las variables de entrada que se tienen. La primera función es para definir el diseño que se va a presentar, la segunda es para

llevar un conteo de los datos que se van a ingresar, la tercera es una función análoga a la segunda, y por último la función encargada de integrar los datos que se tienen en la construcción del adaptador para que estos puedan ser representados de manera correcta en el diseño que se va a elaborar.

El caso del `ViewHolder` es un poco similar al del `ViewPager` puesto que se requiere crear un adaptador para poder conservar los datos.

```
package com.tesis.rudyardb.prueba2

import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import kotlinx.android.synthetic.main.lista.view.*
import java.util.*

class listado(val item : ArrayList<String>, val context: Context, val string: String) : RecyclerView.Adapter<listado.viewHolder>() {

    override fun getItemCount(): Int {...}

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): viewHolder {...}

    override fun onBindViewHolder(holder: viewHolder, position: Int) {...}
    class viewHolder (view: View) : RecyclerView.ViewHolder(view){...}
}
```

Figura 2.13 ViewHolder

Como podemos observar en la figura 2.13, en este caso la función del `ViewHolder` en la primera línea hace exactamente lo mismo que la del `ViewPager`, es un conteo para saber en dónde se encuentra el valor que se desea obtener, la segunda función es donde se define el área de trabajo y a cuál diseño corresponde, por último, la tercera función es la que permite vincular los datos y así poder tenerlos de manera ordenada en una lista. La *clase* `onBindViewHolder` sirve como una forma de poder vincular los diferentes diseños que se tienen; debido a que cada uno de los códigos mostrados en la figura 2.5 corresponden a una sección del código, estos a su vez deben tener su propio diseño para así poder integrarlos en la aplicación de manera adecuada y como es requerido para la presentación de los datos.

CAPÍTULO 3

3. El monitoreo en la nube y porqué usarlo.

En este capítulo se presentará con fundamentos del porqué la solución de monitoreo con dispositivos electrónicos a través de una nube es una manera factible y alterna en comparación con la de asignar cierto número de médicos para el control de varios pacientes. De la misma manera, por medio de cálculos, la localización de los elementos de red, la cobertura esperada y cálculos de tiempos de transmisión y pérdidas de paquetes.

3.1. Fisiología de la red

En esta parte abordaremos la forma en que está constituida la red, de qué manera el medio contribuye en la propagación de las ondas y, además, la posición adecuada para colocar los componentes de red, y de esta manera obtener un correcto funcionamiento que dé cobertura a todas las estaciones móviles que se deseen implementar.

Para poder implementar una red se debe elaborar un estudio de campo y de esta manera, definir algunos aspectos con respecto a la construcción deseada. En estos estudios se suele determinar la posición del equipo transmisor, además de los sitios remotos, los cálculos o estimaciones de pérdidas a considerar por los efectos del medio en el cual viajan las ondas electromagnéticas y, por último, la estimación de la cobertura del área deseada. El área de cobertura comúnmente se la suele graficar por medio de un mapa de calor, esta herramienta visual ayuda a entender la distribución de potencia en una región determinada. En este trabajo se va a presentar un análisis comparativo entre una herramienta para la medición de las potencias de recepción en función de la distancia hasta la cual se tiene cobertura versus un resultado netamente teórico. el cual se procederá a desarrollar en base a los modelos de propagación existentes.

3.2. Cálculo de pérdidas en interiores

Las pérdidas siempre están presentes en las ondas electromagnéticas, independientemente del medio en el que éstas se están propagando. Existen varios modelos (Motley-Cost231, Motley-Simplificado, UIT-R) que sirven para poder determinar de qué manera influyen varios factores ambientales en la potencia que se recibe.

Según el modelo propuesto en [18], es necesario considerar la cantidad de paredes o pisos según sea el caso. La ecuación (3.1) calcula las pérdidas dependiendo del caso.

“Ecuación para encontrar la potencia de recepción conociendo las pérdidas”

$$\begin{aligned} P_r(\mathbf{dB}) = & (P_t + G_t + G_r) - 100.04 + 20 \log(d) \\ & + \sum_{i=1}^P \sum_{j=1}^Q (L_{abs})_{ij} + \sum_{i=1}^P \sum_{j=1}^Q (L_{uabs})_{ij} \end{aligned} \quad (3.1)$$

En ecuación (3.1) podemos ver que la potencia de recepción viene dada por la potencia de transmisión, las ganancias de las antenas y la distancia entre los puntos de transmisión y recepción. También se considera las pérdidas por absorción, multitrayecto y reflexión. Estos cálculos son los teóricos para poder ser objeto de comparación con los resultados experimentales.

Las medidas experimentales se realizaron con una aplicación móvil para medir la potencia de recepción en diferentes distancias, estos datos son presentados en la figura 3.1 para su análisis comparativo.

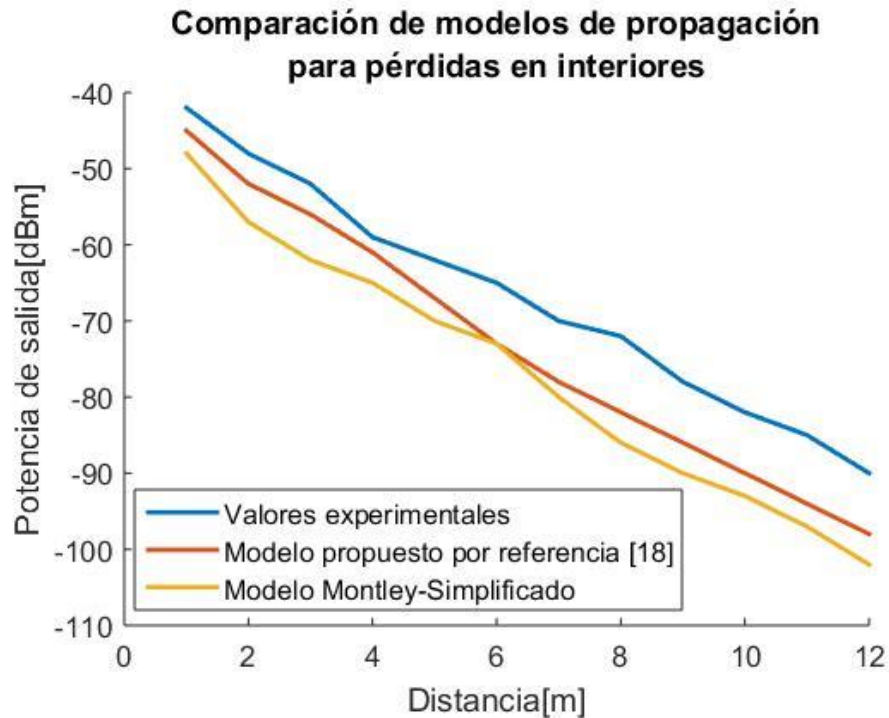


Figura 3.1 Comparación entre el modelo propuesto por [18], modelo Montley-Simplificado [19] y valores experimentales

En la figura 3.1 se muestran las gráficas comparativas de los modelos y el promedio de 20 muestras tomadas por cada metro para poder realizar la gráfica “valores experimentales”, los datos experimentales se obtuvieron por medio de una aplicación celular *Wifi Analyzer*, y los teóricos por medio del modelo propuesto por [18] en la ecuación (3.1). El modelo tiene cierta semejanza con los valores experimentales, es por eso que el porcentaje de error está cerca del 10%, puesto que no son los valores reales sino una aproximación, es probable que sea esta la causa de las variaciones. Sin embargo, podemos decir que es bastante acertado el modelo, considerando que la tasa de error es relativamente baja.

De la ecuación (3.1) podemos decir que, en base a lo obtenido por los cálculos realizados, se puede asegurar que la mejor posición para colocar el punto de red debería ser en el punto medio de las habitaciones, ya que en ese punto las ondas recorren igual distancia entre las habitaciones.

3.3. Desempeño de la red

Una vez que tenemos la estructura física de la red establecida, sus propiedades y su topología, es necesario evaluar cómo la red se va a comportar o desempeñar una vez que pase a ser implementada. La forma en la que se evalúa a la red es con diferentes pruebas para poder medir cómo está trabajando. Entre estas pruebas están las de tiempo de transmisión, interferencia, tasa de pérdidas. Es necesario también realizar una transmisión en tiempo real para analizar si ocurre algún imprevisto o envío no deseado de datos en un determinado tiempo de trabajo.

3.3.1. Tiempo de transmisión de datos

El tiempo de transmisión de datos al que se hace referencia en esta parte es el tiempo de vida media de un paquete o datagrama. El tiempo que un datagrama se demora en poder llegar a su destino sin que ocurra un error o eventualidad, considerando el número de saltos que debe dar antes de ser descartado, es lo más cercano a un tiempo de transmisión que podemos encontrar.

Este valor se puede conocer de varias formas, la que se usó para este trabajo fue enviar un datagrama de prueba para que se pueda conocer el tiempo que le tomó en llegar a su destino. En este caso se procedió a usar la función “*ping*” para poder determinar el tiempo de vida de los datagramas enviados.

```
Attempting to connect to WPA SSID: Rudyarb
You're connected to the network
SSID: Rudyarb
BSSID: 0:1C:10:9D:49:18
signal strength (RSSI): -75
Encryption Type: 2

Pinging www.google.com: SUCCESS! RTT = 90 ms
Pinging www.google.com: SUCCESS! RTT = 80 ms
Pinging www.google.com: SUCCESS! RTT = 90 ms
Pinging www.google.com: SUCCESS! RTT = 110 ms
```

Figura 3.2 Cantidad de tiempo que demora en enviar la información del módulo a la base de datos en tiempo real Firebase

Como vimos en la sección 2.3.3 existe conexión entre la tarjeta y el servidor de Google (Firebase), la cual podemos observar en la figura 3.2 al ver el mensaje que se encuentran conectados, este tiempo que observamos es en realidad el tiempo en que se demora el dispositivo en recibir el aviso de que está conectado a la IP mostrada, entonces podemos decir que en realidad el tiempo de transmisión es 46.25 ms lo que tardan los datos en arribar al servidor.

En términos de calidad, este tiempo es bastante alto, pero no se debe al dispositivo que envía los datos sino a la calidad de la red usada para esto. El tiempo puede variar una vez que se tenga una red dedicada solo a la funcionalidad de los dispositivos. A pesar de esto, el tiempo es lo suficientemente bajo como para considerarse una transmisión en tiempo real. Según la ref. [19], se puede considerar una transmisión en tiempo real aquella que está en valores de hasta 100 ms, y en razón que el tiempo que un doctor toma en visitar al paciente, además de trasladarse a los demás cuartos es mucho mayor que el tiempo que le toma a los componentes captar, transmitir, enviar y recibir los datos según la funcionalidad desempeñada por cada uno de estos dispositivos que forman parte la topología propuesta, podemos decir que el envío de datos es en tiempo real.

3.3.2. Pérdidas de paquetes

Las pérdidas de paquetes, al igual que el tiempo de transmisión, pueden ser obtenidas por medio de la programación de las tarjetas procesadoras de información. Las pérdidas de los paquetes se deben a que, para llegar de un lugar a otro, los datagramas tienen que viajar por diferentes rutas haciendo saltos. Estos saltos al ser varios, producen que ciertos datos que se transmiten se pierdan en el trayecto haciendo posible la propagación de errores a lo largo de la transmisión. Una forma de mitigar esto es usando avisos tanto en la parte que envía como la que recibe. El método para poder tener estos avisos es conocido como Three-way handshake (Protocolo de acuerdo a tres vías) [21], el cual es un método utilizado en

una red TCP (Protocolo de Control de Transmisión) para crear una comunicación entre anfitrión/cliente y un servidor. Este método de tres pasos requiere que el cliente y el servidor intercambian paquetes SYN (bit de control de segmento) y ACK (acuse de recibido) antes de comenzar la comunicación de datos reales.

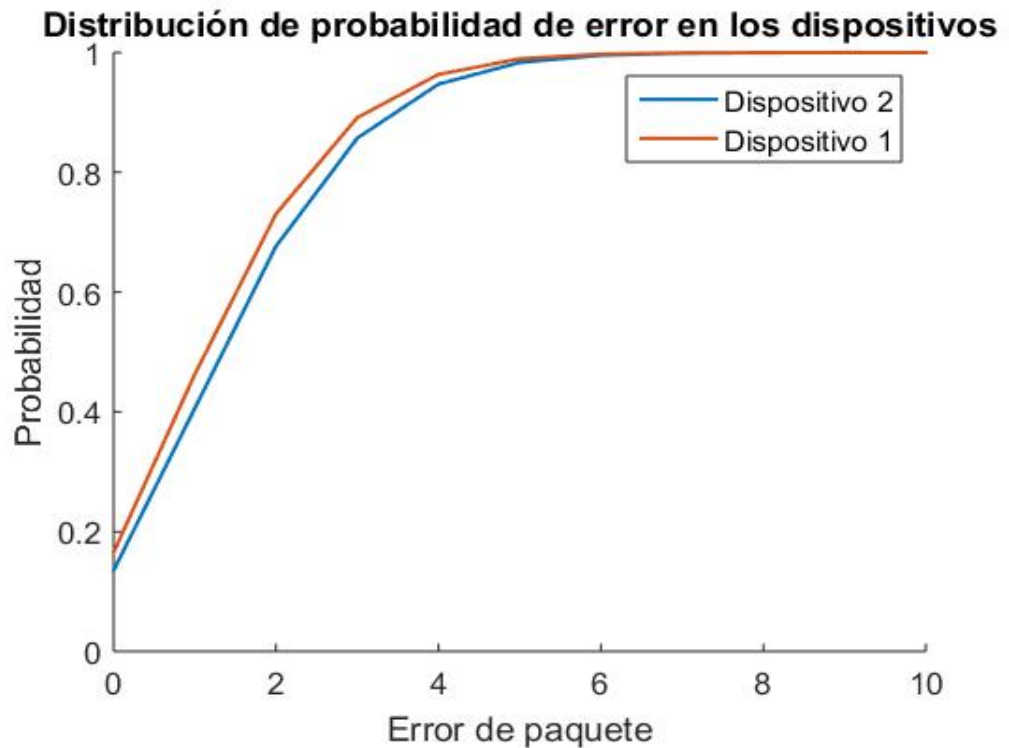


Figura 3.3 Probabilidad de errores producidos en un periodo de 5 horas

Para nuestra implementación, se ha probado la comunicación entre los diferentes equipos en varios escenarios y se pudo apreciar que, si bien es cierto se reciben mensajes de errores cuando no hay conexión a internet, pero los datos nunca se pierden como tal; la base de datos en tiempo real funciona de tal forma que los datos quedan en espera hasta que la conexión sea restablecida y puedan volver a enviarse los datos que se quedaron pendientes de envío. Esto no lo hace exento de que haya paquetes que se pierdan.

Después de tenerlo cinco horas en continuo uso se llegó a la conclusión de que la tasa de paquetes enviados supera el 90% tal como se muestra

en la figura 3.3, en la cual se envió un dato cada 3 minutos durante las 5 horas y solo se presentaron 9 errores en la tarjeta 1 y 10 errores en la tarjeta 2; lo cual lo hace un sistema bastante fiable considerando que es completamente autónomo y que maneja una información bastante importante en el área de interés de este proyecto.

Cabe recalcar que este análisis fue realizado durante 5 horas, pero no significa que los dispositivos no están preparados para un uso continuo.

3.3.3. Interferencia

La interferencia es el fenómeno en el cual las ondas electromagnéticas se superponen en una transmisión, ya sea por el medio en el que se propaga o por intervención de otras ondas que viajan en el medio. Si una onda se propaga en un medio irregular pueden existir problemas como el multitrayecto, reflexión, dispersión y refracción. En caso de que existan más dispositivos presentes en el medio, hay que considerar diferentes factores.

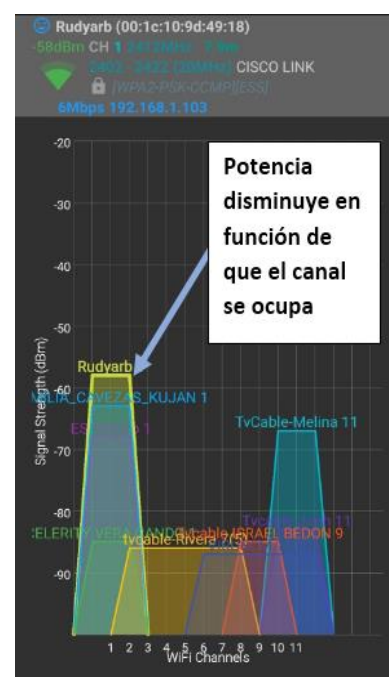
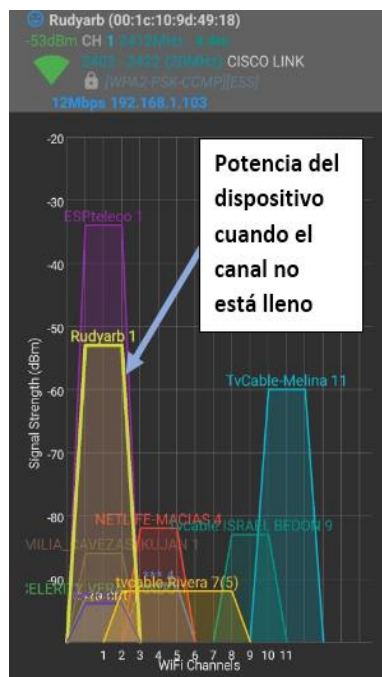
Una onda electromagnética que se propaga con otras, y a la misma frecuencia, es susceptible a que presente interferencias en su recepción; para evitar estos problemas se tienen diferentes formas de transportar estas señales. No solo se presentan problemas si la frecuencia con la que transmiten es la misma, existe una gran posibilidad de que ondas se puedan traslapar entre sí por ocupar parte del canal que le corresponde a otra; estos términos son conocidos como interferencia de canales adyacentes e interferencia de co-canal.

Para poder determinar los problemas a los que el dispositivo puede ser vulnerable, en términos de las interferencias, se dejó funcionando dos dispositivos por un lapso extenso (8 horas) para corroborar que puede trabajar correctamente. Una forma de mitigar estos problemas entre los dispositivos responsables del monitoreo fue definir hacia qué lugar debían apuntar en la base de datos en tiempo real; así como se comentó en la

Sección 2.3.1 la constitución de los datos, recordando que su estructura es como un la de un árbol por lo que se puede aprovechar esta particularidad para construir la ruta adecuada para los datos.

En la parte de la transmisión de los datos se definió que cada dispositivo apunte a un “hijo” diferente de la raíz en la base de datos en tiempo real, haciendo esto nos aseguramos de que no exista problemas de transmisión entre los dispositivos hacia la base de datos.

Con respecto al tiempo de transmisión fue considerados en el cálculo de las pérdidas que se realizó en la Sección 3.2.1. Los problemas de multitrayecto, reflexión y refracción son catalogados como pérdidas en el presupuesto de enlace y esto hace que ya sean considerados como problemas en nuestra transmisión. Los problemas que se experimentaron fueron en las interferencias en el canal que se estaba ocupando, si bien es cierto, la velocidad de transmisión entre estas tarjetas hace posible mitigar los problemas de interferencia con otros dispositivos, no las hace exentas de que enfrenten problemas de utilización de ancho de banda.



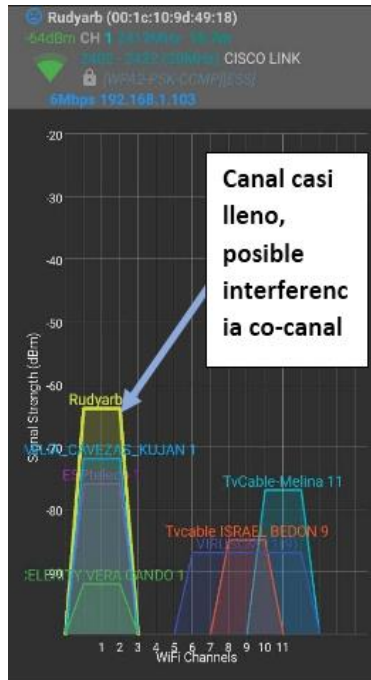


Figura 3.4 Interferencia co-canal producida por dispositivos Wi-Fi

Se hace mención de pérdidas de datos en esta parte puesto que el problema se dio al saturarse el canal en el cual se transmitía, el ancho de banda fue ocupado casi en su totalidad y al no contar con la disponibilidad suficiente se pierden paquetes.

La base de datos en tiempo real no tiene un mecanismo de aviso para que el dispositivo esté al tanto de que se perdieron datos, la forma de corregir esto es que la tarjeta lea lo que transmitió y compare con lo que fue enviado. De esta manera se pudo corregir los errores presentes por este problema de interferencia producido por otros dispositivos.

La tarjeta como tal también presenta problemas de ruido electrónico, este se da en todos los componentes con constante trabajo; al igual que el producido por ocupación del ancho de banda, este ruido electrónico fue corregido comparando los datos que el dispositivo lee y los que envía.

3.4. Costo y Mantenimiento

En esta sección se incluirá el costo de la implementación detallando el valor de cada uno de los elementos que lo conforman y el mantenimiento del mismo para una cama y un cuarto conformado por 8 pacientes.

La limitante de nuestro dispositivo son 20 pacientes, si se desea ampliar el número de pacientes a monitorear será necesario crear una base de datos extra o en su defecto pagar para un mayor almacenamiento en la base de datos.

3.4.1. Costo de la Implementación

El costo de elaboración del Dispositivos Telemédicos para el monitoreo de infantes con síntomas de enfermedades respiratorias como se puede observar en la tabla 4.1 es de \$ 94,00,

Tabla 4.1 Costos de Elaboración Dispositivos monitoreo

Elemento	Unidades	Costo
Router	1	\$ 50,00
Tarjeta Controladora	1	\$ 12,00
Oxipulsímetro	1	\$ 11,00
Termómetro	1	\$ 20,00
Aplicación	1	\$ 1,00
Costo Total		\$ 94,00

Los 8 pacientes con sus respectivas tarjetas controladoras estarán conectados a un mismo routey y una aplicación para que el personal médico realice el monitoreo. El costo total para el monitoreo de pacientes de una sala se detalla en la tabla 4.2

Tabla 4.2 Costos de Elaboración Dispositivos monitoreo para una sala

Elemento	Unidades	Costo por unidad	Total
Router	1	\$ 50,00	\$ 50,00
Tarjeta Controladora	8	\$ 12,00	\$ 96,00
Oxipulsímetro	8	\$ 11,00	\$ 88,00
Termómetro	8	\$ 20,00	\$ 160,00
Aplicación	1	\$ 1,00	\$ 1,00
Costo Total			\$ 395,00

3.4.2. Mantenimiento de la solución propuesta

El proceso de manteniendo se realizará probando el funcionamiento de cada uno de los sensores, así como también la comunicación de la tarjeta con la base de datos. Se realizará pruebas de carga de la batería que se encarga de energizar el módulo.

Se recomienda que el mantenimiento se realice cada 3 meses y si existe alguna duda o se presente algún error informarlo.

3.5. Retroalimentación.

Análisis de los resultados obtenidos en las encuestas.

Las encuestas fueron enviadas y resueltas por personal médico de diferentes áreas de la salud, debido a que, nuestro proyecto se encuentra enfocado en brindar una ayuda a ellos. Obtuvimos respuesta de 24 personas con lo cual procedemos a analizar sus respuestas y con ello validar la utilidad del prototipo.

1. ¿Cuántos pacientes por lo general se encuentran en observación en una sala de hospital?

- a. 2 - 4
- b. 4 - 6
- c. 6 - 8

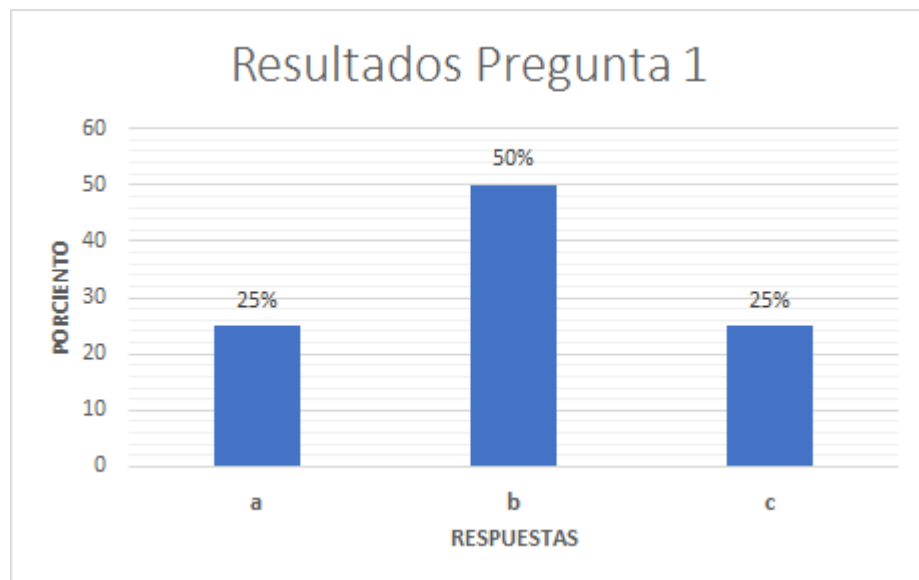


Figura 3.5 Diagrama de barras de la pregunta # 1

En la figura 3.5 podemos apreciar que el 25% (6 de 24) de las personas encuestadas se han encontrado con cuarto que están de 2 a 4 pacientes,

el 25 % (6 de 24) personas encuestadas se han encontrado con 4 a 6 pacientes en un cuarto y el 50% (12 de 24) personas encuestadas se encuentran con cuartos de hospital con 6 a 8 personas, lo que nos muestra que este proyecto es de gran utilidad al momento de querer conocer o monitorear el estado de los pacientes en un cuarto de hospital, debido a que estos pacientes con enfermedades respiratorias el estado de salud tienden a cambiar en cualquier momento.

2. Considera necesario un sistema que le brinde la posibilidad de monitorear los signos vitales de pacientes de una sala de hospital, desde cualquier lugar (oficina, pasillo, consultorio, etc.)

- a. Si
- b. No
- c. Tal vez

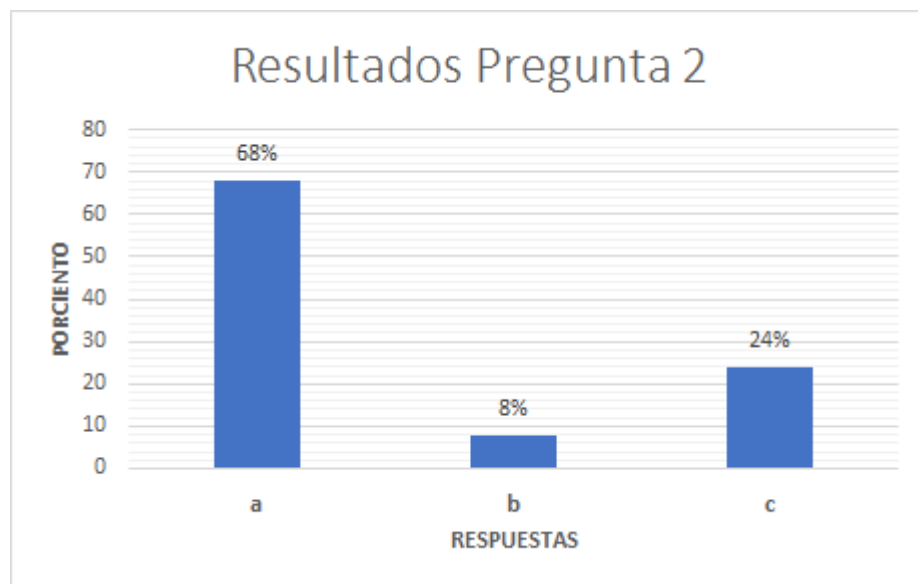


Figura 3.6 Diagrama de barras de la pregunta # 2

Como podemos observar en la figura 3.6, el 8% (2 de 24) de las personas encuestadas consideran que no sería necesario, el 24% (6 de 24) de las personas creen que tal vez sería necesario contar con esta ayuda y 68% (16 de 24) de las personas encuestadas consideran que un sistema que les permita monitorear el estado de los pacientes en el un cuarto de hospital es necesario, esto puede ser debido a que se han encontrado con situaciones en la cual el hospital se encuentra con mucha demanda y el

personal médico no se da basto para realizar las rondas por cada cuarto. Estas personas ven con este proyecto la posibilidad de que exista una gran ayuda al momento de tener muchos pacientes en el cuarto de observación.

3. La visualización de los signos vitales en la aplicación es:

- a. Fácil de observar.
- b. No se puede observar.
- c. Necesita alguna mejora

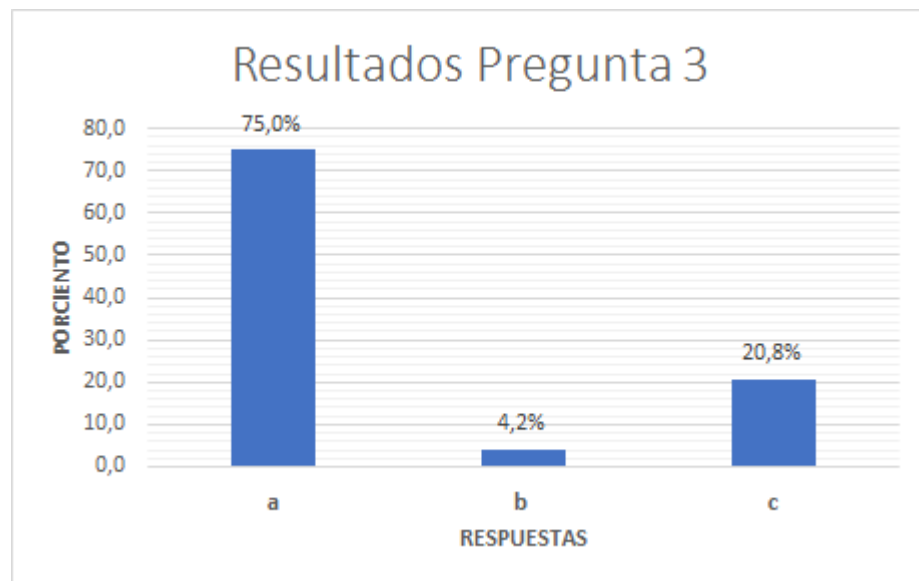


Figura 3.7 Diagrama de barras de la pregunta # 3

Con la finalidad de tener una retroalimentación a través de las personas encuestadas decidimos colocar en la pregunta número 3 una foto de la captura de pantalla del celular (esta imagen se encontrará en la sección Anexo A). Como se observa en la figura 3.7, obtuvimos que el 4.2% (1 de 24) de las personas encuestadas considera que no se puede observar la información provista por la aplicación, el 20.8% (5 de 24) de los encuestados consideran que la aplicación necesita alguna mejora y 75% (18 de 24) de los encuestados creen que es fácil observar la información en la aplicación.

Nos contactamos con los encuestados para conocer qué les gustaría mejorar en cuanto a la visualización en la aplicación y nos comentaron que estaría bien agrandar un poco el tamaño de la letra y a su vez tabular la información.

4. ¿Cuál es el aspecto que considera más importante de la aplicación?

a. Puede ser utilizados en cualquier dispositivo móvil con acceso a internet.

b. La información obtenida es visualizada en tiempo real.

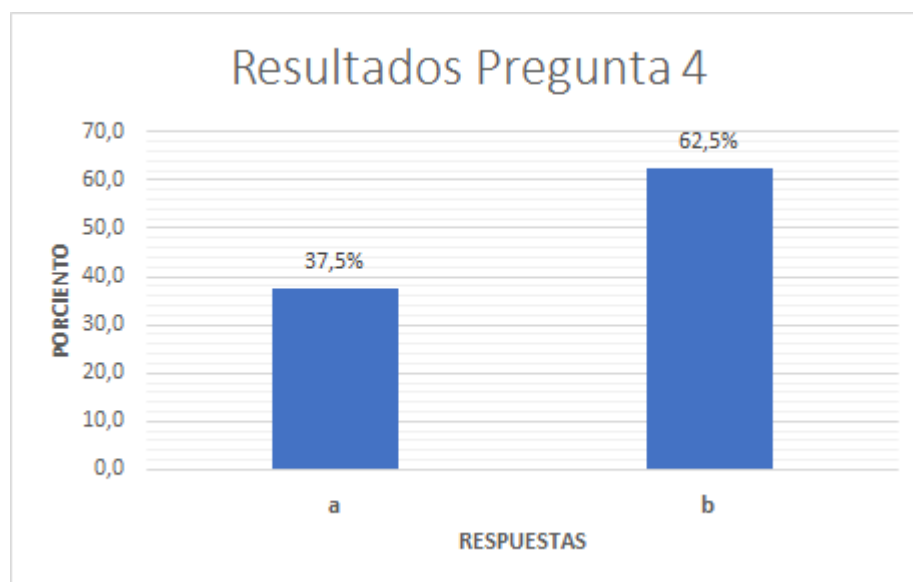


Figura 3.8 Diagrama de barras de la pregunta # 4

Como podemos observar en la figura 3.8 el 37.5% (9 de 24) de los encuestados consideran que el aspecto más importante en que puede ser utilizados en cualquier dispositivo móvil con acceso a internet, y el 62.5% (15 de 24) creen que lo más importante de la aplicación es que la información obtenida es visualizada en tiempo real

Luego de analizar las respuestas que se lograron obtener de cada una de las preguntas de la encuesta podemos mencionar que se cumplió con el cometido de esta, el cual era corroborar la aceptación y utilidad que tendrá nuestro prototipo.

Para realizar esta encuesta se utilizó la herramienta de formularios de *Google* la cual nos permitió obtener más rápido las respuestas de los encuestados, así como también el diagrama de barras para el análisis de cada pregunta del formulario.

Capítulo 4

4. Conclusiones y Recomendaciones

4.1. Conclusiones

Dado el medio en el cual se tomaron las medidas de potencia, según lo comparado con el modelo de la referencia [18], se puede apreciar que es una aproximación válida para el entorno en el cual se desea poner la red. En el modelo montley se tiene en consideración la cantidad de pisos y paredes que están en el ambiente el cual se propagaran las ondas, pero en la referencia [18] además de considerar la estructura y el material que las componen, se tiene que intervienen más factores como lo son: las perdidas multitrayecto, las perdidas por refracción y reflexión y perdidas por desvanecimiento.

Como se puede ver en lo expuesto en el capítulo anterior, existen varios factores que pueden degradar la calidad de la trasmisión tales como interferencias de co-canal o posiblemente de canales adyacentes, debido a la velocidad en la que se transmiten los datos internamente en el dispositivo electrónico que es de 9600 baudios; se puede evitar que exista algún tipo de interferencia que envié datos erróneos a la base de datos. Si bien es cierto esto ayuda a que no se envíen datos incorrectos aún existe la posibilidad de que haya demora en la trasmisión de los datos por problemas de interferencia co-canal.

Se puede visualizar que el tiempo de transmisión de los datos que son adquiridos por el dispositivo electrónico hasta la aplicación móvil es de 50 ms aproximadamente. Según la referencia [19] se puede considerar una transferencia en tiempo real a aquellas cuyo tiempo de transmisión es menor a 100 ms, dado que en las pruebas de envío realizadas, si el sistema tiene una conexión a internet dedicada solo para la trasmisión de los datos se llega a cumplir con lo mencionado en el trabajo [19].

4.2. Recomendaciones

La aplicación debe ser mejorada en términos de estética y visualización, debido a que, según algunos encuestados, la visualización no era la adecuada para percibir los datos mostrados, adicional a esto se puede agregar una función que permita al usuario autenticarse al momento de ingresar ya que esto la haría más segura y restringida.

Como trabajo futuro se puede considerar que el dispositivo pueda monitorear a los pacientes que estén fuera del hospital, esto permite añadir movilidad al usuario sin sacrificar la ventaja de que el personal médico pueda acceder a la información donde quiera que esté el paciente. Se debe considerar ofrecer un punto de acceso a internet solo para los dispositivos médicos, así se puede evitar problemas de interferencia co-canal y demás.

Para trabajo futuro se puede tener una base de datos de enfermedades que se pueden determinar por medio de la observación de los signos vitales fuera del rango normal. Es decir que se pueda determinar una enfermedad aproximada conociendo los signos vitales.

Se recomienda tener una red dedicada para el sistema telemédico para evitar que existan errores al enviar la información a la base de datos de *Firebase* y por ende evitar pérdida de información al visualizar la aplicación.

BIBLIOGRAFÍA

- [1] World Health Organization, (2016). Rate of deaths in the world
en: <http://www.who.int/en/news-room/fact-sheets/detail/the-top-10-causes-of-death>
- [2] Organización Panamericana de Salud, (2014). Principales causas de muerte para los países y territorios de las Americas: Sub región andina.
en:<http://www.paho.org/data/index.php/es/mnu-mortalidad/principales-causas-de-muerte.html?showall=&start=2>
- [3] Organización Panamericana de Salud, (2014). Principales causas de muerte en el Ecuador.
en:<http://www.paho.org/data/index.php/es/mnu-mortalidad/principales-causas-de-muerte.html?showall=&start=2>
- [4] World Health Organization, (2018, Febrero). Rate of deaths by country
en: <http://apps.who.int/gho/data/view.main.ghe2002015-ECU>
- [5] Número de habitantes del Ecuador,(2016).
en:<https://www.datosmacro.com/demografia/poblacion/ecuador>
- [6] G. Iugmaña, “Anuario de estadísticas hospitalarias, camas y egresos 2014”, Instituto Ecuatoriano de Estadísticas y Censos, Ecuador,2014.
- [7] B. Crow, I. Widjaja, J. Kim, P. Sakai “IEEE 802.11 Wireless Local Area Networks” IEEE Communications Magazine, vol 35, Issue 9, September 1997.
- [8] M. Jung, C. Reinisch, W. Kastner, “Integrating Building Automation Systems and IPv6 in the Internet of Things”, en Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Palermo, Italy, 2012, pp 683-688
- [9] S. Kelly, N. Suryadevara, S. Mukhopadhyay, “Towards the Implementation of IoT for Environmental Condition Monitoring in Homes”, IEEE SENSORS JOURNAL, vol 13, No, 10, October 2013
- [10] L. Yu, Y. Yu, X. Zhu,” Smart Hospital base on Internet of Things” Journal of networks, vol 7, NO. 10, October 2012.
- [11] D. Lu y T. Liu,” The Application of IOT in Medical System”, en IT in Medicine and Education (ITME) International Symposium, Cuangzhou, China, 2011, pp. 272-275.
- [12] I. Chiuchisan, H. Costin, O. Geman,” Adopting the Internet of Things Technologies in Health Care Systems”,en International Conference and Exposition on Electrical and Power Engineering (EPE 2014) ,Iasi, Romania, 2014, pp. 532-535.
- [13] M. Uenoyama, T. Matsui, K. Yamada, S. Suzuki, B. Takase, S. Suzuki, M. Ishihara, M. Kawakami,”Non-contact respiratory monitoring system using a ceiling-attached microwave antenna”, Med Bio Eng Comput (2006) ,Tokyo, Japón, ©International Federation for Medical and Biological Engineering, DOI:10.1007/s11517-006-0091-8.

- [14] G. Yang, L. Xie, M. Mäntysalo, X. Zhou, Z. Pang, L. Da Xu, S. Kao-Walter, Q. Chen, L. Zheng, "A Health-IoT Platform Based on the Integration of Intelligent Packaging, Unobtrusive Bio-Sensor and Intelligent Medicine Box", IEEE Transactions on Industrial Informatics, Vol 10, Issue 4, November 2014
- [15] Melexis: <https://www.melexis.com/en/product/MLX90614/Digital-Plug-Play-Infrared-Thermometer-TO-Can>
- [16] Ventura, V. (9 de Noviembre de 2016). *Polaridad*. Obtenido de <https://polaridad.es/max30100-sensor-latido-corazon-oximetro-pulso-i2c-wearable-salud/>
- [17] Llamas, L. (29 de Octubre de 2016). luis llamas. Obtenido de <https://www.luisllamas.es/arduino-y-el-termometro-infrarrojo-a-distancia-mlx90614/>
- [18] P. Thu Zar Tun, A. Su Hlaing, "Analysing Radio Wave Propagation Model for Indoor Wireless Communication", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 4, April 2013
- [19] Kaijun Fan, Bingyin Xu, Guofang Zhu and Jie Gao, "Fast peer-to-peer real-time data transmission for distributed control of distribution network," *2014 China International Conference on Electricity Distribution (CICED)*, Shenzhen, 2014, pp. 1041-1045.
doi: 10.1109/CICED.2014.6991864
- [20] Sandobal, M. F. (3 de Julio de 2014). SLIDESHARE. Obtenido de https://es.slideshare.net/blog_fralbe/modelos-de-propagacin-interiores
- [21] techopedia. (29 de 10 de 2018). *TECHOPEDIA*. Obtenido de <https://www.techopedia.com/definition/10339/three-way-handshake>

ANEXOS

Anexo A

Formato del formulario utilizado para las encuestas.

A continuación mostramos la encuesta realizada a través de Formularios de Google.

PREGUNTAS RESPUESTAS 25

Sistema para monitoreo de signos vitales de Pacientes con enfermedades respiratorias

Este formulario fue elaborado con la finalidad de conocer el nivel de aceptación de nuestro producto.

1. ¿Cuántos pacientes por lo general se encuentran en observación en una sala de hospital? *

2-4

4-6

6-8

2. Considera necesario un sistema que le brinde la posibilidad de monitorear los signos vitales de pacientes de una sala de hospital, desde cualquier lugar (oficina, pasillo, consultorio, etc) *

Sí

No

Tal vez

3. La visualización de los signos vitales en la aplicación es: *



- Fácil de observar.
- No se puede observar.
- Necesita alguna mejora

4. ¿Cuál es el aspecto que considera más importante de la aplicación? *

- Puede ser utilizados en cualquier dispositivo móvil con acceso a internet.
- La información obtenida es visualizada en tiempo real.

Anexo B

Presentación de datos en aplicación móvil.



Capturas de la aplicación

Anexo C

Manual de usuario.

Para el uso de la red tomar en cuenta las siguientes instrucciones una vez que se haya adquirido el servicio.

1. Colocar los sensores en el paciente según el médico lo considere.
2. Encender el dispositivo electrónico encargado de adquirir los datos.
3. Ingresar a la plataforma de datos en tiempo real con el usuario y contraseña asignada.
4. Una vez que se completaron los pasos 2 y 3 verificar en la base de datos si se está recibiendo información.
5. Abrir la aplicación en el dispositivo móvil.
6. Para verificar los datos de cada paciente es necesario solo deslizar la pantalla.

Se recomienda revisar el estado del dispositivo cada hora para verificar si existe algún sobrecalentamiento, además es importante tener cuidado con la manipulación de los sensores puesto que estos pueden averiarse y mandar datos erróneos.

Anexo D

Código de programación.

Programación del dispositivo electrónico

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

// Set these to run example.
#define FIREBASE_HOST "prueba-8415d.firebaseio.com"
#define FIREBASE_AUTH "kMxR8UsrFrLrBvTFRkeGX5cbhW37hobhty8S3M4A"
#define WIFI_SSID "Z3"
#define WIFI_PASSWORD "aaaaaaaa"

void setup() {
  Serial.begin(9600);
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
  int n=random(70,100);
  int a=random(80,150);
  int b=random(60,100);
  int t=random(37,42);
  // append a new value to /logs
  //String name = Firebase.pushString("Paciente 1");
  if(n<90){
    Firebase.setInt("Paciente 1/SO2",n);
    delay(1500);
  }
  else{
    Firebase.setString("Paciente 1/SO2","alarma");
    delay(1500);
  }
  if(a>100 && a<120){
```

```

    Firebase.setInt("Paciente 1/P_A SIST",a);
    delay(1500);
}
else{
    Firebase.setString("Paciente 1/P_A SIST","alarma");
    delay(1500);
}
if(b>85 && b<95){
    Firebase.setInt("Paciente 1/P_A DIAST",b);
    delay(1500);
}
else{
    Firebase.setString("Paciente 1/P_A DIAST","alarma");
    delay(1500);
}
if(t>37 && t<40){
    Firebase.setInt("Paciente 1/TEMPERATURA",t);
    delay(1500);
}
else{
    Firebase.setString("Paciente 1/TEMPERATURA","alarma");
    delay(1500);
}
// handle error
if (Firebase.failed()) {
    Serial.print("pushing /logs failed:");
    Serial.println(Firebase.error());
    return;
}
Serial.print("pushed: /logs/");
delay(1000);
}

```

Programación de aplicación móvil

Página principal

```
package com.tesis.rudyarb.prueba2
```

```

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.google.firebase.database.*
import kotlinx.android.synthetic.main.activity_main.*

```

```

class MainActivity : AppCompatActivity() {
    class Paciente(var paciente: String, var sintomas: String) {
        override fun equals(other: Any?): Boolean {
            if (this === other) return true
            if (javaClass != other?.javaClass) return false

            other as Paciente

```

```

        if (paciente != other.paciente) return false

        return true
    }

    override fun hashCode(): Int {
        return paciente.hashCode()
    }
}

var images: Array<Int> = arrayOf(R.drawable.imagenr1, R.drawable.imagenr1)
var listas: ArrayList<Paciente> = ArrayList()

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    val refer = FirebaseDatabase.getInstance()
    val abc = refer.getReference()
    var adapter = adapter(images, applicationContext, listas)
    view_pager.adapter = adapter
    abc.addValueEventListener(object : ValueEventListener {

        override fun onCancelled(p0: DatabaseError) {

        }

        override fun onDataChange(snapshot: DataSnapshot) {
            val elements = snapshot.children.toMutableList()
            for (element in elements) {
                var b = Paciente(element.key!!, element.toString())
                var a = listas.indexOf(b)
                if (a == -1) {
                    listas.add(b)

                } else
                    listas[a].sintomas = element.toString()
            }
            adapter.notifyDataSetChanged()
        }
    })
}
}
}

```

Programación de adaptador

```
package com.tesis.rudyarb.prueba2
```

```
import android.content.Context
```



```
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import androidx.viewpager.widget.PagerAdapter
import android.widget.RelativeLayout
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
```

```
class adapter: PagerAdapter {
    var context: Context
    var images: Array<Int>
    var paciente: ArrayList<MainActivity.Paciente>
```

```
    lateinit var inflater: LayoutInflater
```

```
    constructor(images: Array<Int>, context: Context, paciente:
ArrayList<MainActivity.Paciente>) : super() {
        this.context = context
        this.images = images
        this.paciente = paciente
    }
```

```
    override fun isViewFromObject(view: View, `object`: Any): Boolean = view ==
`object` as RelativeLayout
```

```
    override fun getCount(): Int = paciente.size
```

```
    override fun getItemPosition(`object`: Any): Int = images.size
```

```
    override fun instantiateItem(container: ViewGroup, position: Int): Any {
        val image: ImageView
        val lista: RecyclerView
```

```
        inflater = context.getSystemService(Context.LAYOUT_INFLATER_SERVICE) as
LayoutInflater
```

```
        val view: View = inflater.inflate(R.layout.image_item, container, false)
        image = view.findViewById(R.id.slider_image)
        lista = view.findViewById(R.id.listapaciente)
        lista.layoutManager = LinearLayoutManager(context)
```

```
        lista.adapter = listado(arrayListOf(paciente[position].sintomas), context,
paciente.get(position).sintomas)
        container.addView(view)
        return (view)
    }
}
```

Anexo E

Detalles de Sensores y Tarjeta.

Termómetro infrarrojo digital sin contacto (MLX90614)

El Sensor de Temperatura infrarrojo MLX90614 es un chip de silicio con una fina membrana micromecanizada, diseñada para ser sensible a la radiación infrarroja emitida por un objeto a distancia. El sensor posee una etapa de amplificación y digitalización de la señal procedente de la membrana. La salida del sensor es lineal y se compensa de acuerdo con las variaciones de la temperatura ambiente.

El sensor MLX90614 integra un circuito de filtrado de ruido, un conversor A/D de 17 bits de resolución y un procesador digital de señales, entregando un amplio rango de trabajo para objetos desde -70°C hasta 380°C , con una precisión de 0.5°C .

La salida del sensor es de tipo SMBus, que es muy similar al protocolo I2C, además se puede configurar una salida PWM de 10 bits.

Oxímetro de pulso (MAX30100)

La pulsioximetría es un método no invasivo, que permite medir el porcentaje de saturación de oxígeno de la hemoglobina (SaO_2) en sangre de un paciente con utilizando un circuito de fotoeléctricos. Para esto se emplea un pulsioxímetro, que es un dispositivo que integra los emisores de luz y el sensor que mide la cantidad de luz reflejada por el dedo del paciente. La luz detectada por el sensor varía de acuerdo a la concentración de oxígeno en la sangre, la sangre oxigenada absorbe mayor cantidad de luz infrarroja, mientras que la sangre poco oxigenada absorbe mayor luz roja.

El MAX30100 es un dispositivo que integra un pulsioxímetro y un monitor de frecuencia cardiaca. Posee dos Leds: un led rojo (660nm) y un led infrarrojo (920nm), un fotodetector, óptica especializada, filtro de luz ambiental entre 50 y 60Hz, y un conversor ADC delta sigma de 16 bits y de hasta 1000 muestras por segundo. Además posee un sensor de temperatura interno para compensar los efectos de la temperatura en la medición.

El MAX30100 necesita de dos voltajes para funcionar: 1.8V y 3.3V, por lo que este módulo incluye ambos reguladores de voltaje en placa, de ese modo solo se necesita una fuente de 5V para la alimentación. Su consumo de corriente es mínimo, por lo que es ideal para aplicaciones portátiles. Puede ser utilizado en equipos de monitoreo médico, asistentes de estado físico.

Tarjeta controladora Hiletgo

NodeMCU es una tarjeta de desarrollo similar a Arduino, especialmente orientada al Internet de las cosas (IoT). Está basado en el SoC (System on Chip) ESP8266, un chip altamente integrado, diseñado para las necesidades de un mundo conectado. Integra un potente procesador con Arquitectura de 32 bits (más potente que el Arduino Due) y conectividad Wifi.

Para el desarrollo de aplicaciones se puede elegir entre los lenguajes Arduino y Lua. Al trabajar dentro del entorno Arduino podremos utilizar un lenguaje que ya conocemos y hacer uso de un IDE sencillo de utilizar, además de hacer uso de toda la información sobre proyectos y librerías disponibles en internet. La comunidad de usuarios de Arduino es muy activa y da soporte a plataformas como el ESP8266.

NodeMCU viene con un firmware pre-instalado el cual nos permite trabajar con el lenguaje interpretado LUA, enviándole comandos mediante el puerto serial (CP2102). Las tarjetas NodeMCU y Wemos D1 mini son las plataformas más usadas en proyectos de Internet de las cosas (IoT). No compite con Arduino, pues cubren objetivos distintos, incluso es posible programar NodeMCU desde el IDE de Arduino.

La tarjeta NodeMCU está diseñada especialmente para trabajar en protoboard. Posee un regulador de voltaje en placa que le permite alimentarse directamente del puerto USB. Los pines de entradas/salidas trabajan a 3.3V. El chip CP2102 se encarga de la comunicación USB-Serial.

Especificaciones Técnicas.

Voltaje de Alimentación (USB): 5V DC

Voltaje de Entradas/Salidas: 3.3V DC

SoC: ESP8266 (Módulo ESP-12)

CPU: Tensilica Xtensa LX3 (32 bit)

Frecuencia de Reloj: 80MHz/160MHz

Instruction RAM: 32KB

Data RAM: 96KB

Memoria Flash Externa: 4MB

Pines Digitales GPIO: 17 (pueden configurarse como PWM a 3.3V)

Pin Analógico ADC: 1 (0-1V)

UART: 2

Chip USB-Serial: CP2102

Certificación FCC

Antena en PCB

802.11 b/g/n

Wi-Fi Direct (P2P), soft-AP

Stack de Protocolo TCP/IP integrado

PLLs, reguladores, DCXO y manejo de poder integrados

Potencia de salida de +19.5dBm en modo 802.11b

Corriente de fuga menor a 10uA

STBC, 1x1 MIMO, 2x1 MIMO

A-MPDU & A-MSDU aggregation & 0.4ms guard interval

Wake up and transmit packets in < 2ms

Consumo de potencia Standby < 1.0mW (DTIM3)