



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

Tema:

**SISTEMA ELECTRÓNICO DE MONITOREO DE MASCOTAS PARA LA
GESTIÓN DE CLÍNICAS VETERINARIAS UTILIZANDO VOIP E IOT.**

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniera en Electrónica y Comunicaciones.

ÁREA: Electrónica

LÍNEA DE INVESTIGACIÓN: Tecnología de la información y Sistema de control.

AUTORA: Yulissa Cecibel León Troya.

TUTOR: Ing. Víctor Santiago Manzano Villafuerte, Mg.

Ambato – Ecuador

marzo - 2022

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: SISTEMA ELECTRÓNICO DE MONITOREO DE MASCOTAS PARA LA GESTIÓN DE CLÍNICAS VETERINARIAS UTILIZANDO VOIP E IOT, desarrollado bajo la modalidad Proyecto de Investigación por la señorita YULISSA CECIBEL LEÓN TROYA, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, marzo 2022.

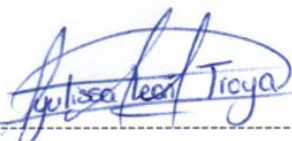
Ing. Víctor Santiago Manzano Villafuerte, Mg.

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: SISTEMA ELECTRÓNICO DE MONITOREO DE MASCOTAS PARA LA GESTIÓN DE CLÍNICAS VETERINARIAS UTILIZANDO VOIP E IOT es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2022.



Yulissa Cecibel León Troya

C.C. 0503728016

AUTORA

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por la señorita Yulissa Cecibel León Troya, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA ELECTRÓNICO DE MONITOREO DE MASCOTAS PARA LA GESTIÓN DE CLÍNICAS VETERINARIAS UTILIZANDO VOIP E IOT, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, marzo 2022.

Ing. Pilar Urrutia, Mg.

PRESIDENTA DEL TRIBUNAL

Ing. Marco Jurado
PROFESOR CALIFICADOR

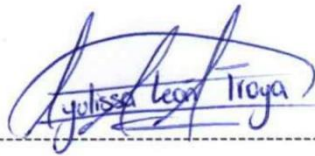
Ing. Juan Pablo Pallo
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2022.



Yulissa Cecibel León Troya

C.C. 0503728016

AUTORA

DEDICATORIA

El presente trabajo está dedicado primeramente a Dios ya que gracias a toda la fe que he puesto en él he culminado esta etapa de mi vida.

A mis padres (Marco y Yolanda), que siempre me brindaron su apoyo y consejos para ser de mí una mejor persona y que pese a las dificultades siempre estuvieron presentes.

A mis hermanos (Fabricio y Jefferson), que han sido siempre un ejemplo para poder alcanzar mis sueños.

A mi abuelita Esther que desde el cielo me cuida para poder cumplir cada meta que me propongo.

A mi primer sobrino que está en camino por ser el medio de alegría y motivación para seguir adelante.

Al amor que Dios puso en mi camino, tu ayuda ha sido fundamental, has estado siempre motivándome y ayudándome hasta donde tus alcances lo permitían.

Yulissa León

AGRADECIMIENTO

A mis padres, hermanos y cuñadas que siempre estuvieron con un consejo o una palabra de aliento gracias por nunca dejar que me rindiera.

A mis compañeros y amigos quienes con sus ocurrencias y apoyo fueron la manera divertida de ver las adversidades de la vida.

A la Universidad Técnica de Ambato que fue mi hogar durante mi formación profesional.

Al Ingeniero Santiago Manzano que con sus conocimientos y ayuda incondicional durante la realización de este trabajo supo ser un ejemplo de excelente persona y gran profesional.

Y a mis amuletos de la buena suerte Sasy y Kuky, mis compañeras fieles que durante las noches de desvelo siempre estuvieron junto a mí.

INDICEGENERAL

| | |
|---|-----|
| PORTADA..... | i |
| APROBACIÓN DEL TUTOR..... | ii |
| AUTORÍA | iii |
| APROBACIÓN TRIBUNAL DE GRADO..... | iv |
| DERECHOS DE AUTOR | v |
| DEDICATORIA | vi |
| AGRADECIMIENTO | vii |
| INDICE DE FIGURAS | xi |
| INDICE DE TABLAS | xiv |
| RESUMEN EJECUTIVO | xv |
| ABSTRACT..... | xvi |
| CAPÍTULO I | 1 |
| MARCO TEÓRICO | 1 |
| 1.1 Antecedentes Investigativos | 1 |
| 1.2 Contextualización del problema..... | 8 |
| 1.3 Fundamentación Teórica | 10 |
| 1.3.1 Salud en mascotas..... | 10 |
| 1.3.2 Sanidad móvil..... | 10 |
| 1.3.3 Entorno de desarrollo..... | 11 |
| 1.3.3.1 React | 11 |
| 1.3.3.2 React Native..... | 11 |
| 1.3.4 Renderizado | 13 |
| 1.3.5 Internet de las cosas..... | 13 |
| 1.3.5.1 Componentes IoT..... | 14 |
| 1.3.6 Software de gestión de la nube | 16 |
| 1.3.7 Tecnologías de la información aplicadas en e-Salud..... | 17 |
| 1.3.8 Enfermedades Cardíacas en animales domésticos..... | 21 |

| | |
|---|-----------|
| 1.3.8.1 Diagnóstico de cardiopatías en perros | 22 |
| 1.3.8.2 Frecuencia cardíaca en perros..... | 23 |
| 1.3.8.3 Saturación de oxígeno en perros (SpO2)..... | 24 |
| 1.3.9 Adquisición y monitoreo de los signos vitales | 24 |
| 1.3.10 Placa de Desarrollo..... | 25 |
| 1.4 Objetivos..... | 30 |
| 1.4.1 Objetivo General..... | 30 |
| 1.4.2 Objetivos Específicos | 30 |
| CAPÍTULO II | 31 |
| METODOLOGÍA | 31 |
| 2.1 Materiales | 31 |
| 2.1.1 Encuestas | 31 |
| 2.2 Métodos | 35 |
| 2.2.1 Modalidad de investigación | 35 |
| 2.2.2 Recolección de información..... | 36 |
| 2.2.3 Procesamiento y análisis de datos | 36 |
| 2.2.4 Propuesta de Solución | 37 |
| 2.2.5 Desarrollo del Proyecto..... | 38 |
| CAPÍTULO III | 39 |
| RESULTADOS Y DISCUSIÓN | 39 |
| 3.1. Análisis y discusión de los resultados | 39 |
| 3.2 Análisis de Factibilidad..... | 39 |
| 3.3 Desarrollo de la propuesta..... | 40 |
| 3.3.1 Adquisición y transmisión de signos vitales..... | 41 |
| 3.3.2 Gestión de información y monitoreo de datos | 47 |
| 3.4 Implementación final del proyecto | 61 |
| 3.5 Verificación de hipótesis | 70 |
| 3.5.1 Pruebas de funcionamiento..... | 70 |
| 3.5.2 Costo del prototipo Precios de Hardware..... | 86 |
| 3.5.3 Resultados | 89 |
| CAPÍTULO IV | 92 |

| | |
|--------------------------------------|----|
| CONCLUSIONES Y RECOMENDACIONES | 92 |
| 4.1 Conclusiones | 92 |
| 4.2 Recomendaciones | 93 |
| BIBLIOGRAFÍA | 94 |
| ANEXOS | 97 |

INDICE DE FIGURAS

| | |
|---|----|
| Figura 1: Diagramas de bloque del sistema experto en línea | 2 |
| Figura 2: Sistema THT | 3 |
| Figura 3: Analizador del sistema THT | 4 |
| Figura 4: Diagrama Esquemático | 5 |
| Figura 5: Estrategias multiobjetivo de programación..... | 6 |
| Figura 6: Descripción general del sistema de diseño de collar inteligente | 7 |
| Figura 7: Renderizado ReactJS vs React Native. | 13 |
| Figura 8: Arquitectura IoT..... | 14 |
| Figura 9: Diagnóstico de cardiopatía por auscultación..... | 22 |
| Figura 10: Sensor MAX30100..... | 28 |
| Figura 11: Sensor NJL5501R. | 28 |
| Figura 12: Sensor SFH7050..... | 29 |
| Figura 13: Pantalla OLED 0.91”..... | 30 |
| Figura 14: Médicos veterinarios encuestados..... | 32 |
| Figura 15: Diagrama de barras resultados primera pregunta. | 32 |
| Figura 16: Diagrama de barras resultados segunda pregunta. | 33 |
| Figura 17: Diagrama de barras resultados tercera pregunta..... | 34 |
| Figura 18: Diagrama de barras resultados cuarta pregunta..... | 34 |
| Figura 19: Estructura general del sistema de medición de signos vitales..... | 40 |
| Figura 20: Estructura del sistema de adquisición y transmisión de datos | 41 |
| Figura 21: Código de librerías utilizadas en la programación Arduino IDE. | 42 |
| Figura 22: Principio de funcionamiento sensor MAX30100..... | 43 |
| Figura 23: Código de acondicionamiento sensor MAX30100. | 44 |

| | |
|--|----|
| Figura 24: Diagrama de conexión de los elementos de la pechera portable. | 45 |
| Figura 25: Diagrama de conexión para el suministro de energía del circuito. | 46 |
| Figura 26: Creación de la aplicación en React Native. | 47 |
| Figura 27: Programación del Splash de bienvenida en React Native. | 48 |
| Figura 28: Página de inicio principal programado en React Native. | 48 |
| Figura 29: Pantallas de inicio de sesión para médico y clientes. | 49 |
| Figura 30: Consulta a la base de datos mediante Express y logeo exitoso. | 50 |
| Figura 31: Creación de la aplicación en React Native. | 50 |
| Figura 32: Pantalla de consulta de pacientes en React Native. | 51 |
| Figura 33: Pantalla de registro de pacientes en React Native. | 51 |
| Figura 34: Pantalla de ingreso de datos de vacunación en React Native. | 52 |
| Figura 35: Pantalla de bienvenida al usuario en React Native. | 53 |
| Figura 36: Carnet virtual en React Native. | 53 |
| Figura 37: Lista de dispositivos bluetooth vinculados. | 54 |
| Figura 38: Pantalla de conexión a dispositivo Bluetooth. | 54 |
| Figura 39: Código de lectura de datos del bus serial del módulo HC-05 | 55 |
| Figura 40: Código de insertar datos en base de datos y visualización en pantalla. | 55 |
| Figura 41: Componentes del servidor en la nube | 56 |
| Figura 42: Diagrama UML de Base de datos. | 57 |
| Figura 43: Estructura del sistema de teleasistencia. | 59 |
| Figura 44: Ingreso a la instancia de Centos 7. | 60 |
| Figura 45: Circuito impreso y su case de protección. | 61 |
| Figura 46: Pechera portable ensamblada. | 62 |
| Figura 47: Base de datos creadas en el sistema y tablas de ‘Almacenamiento’. | 63 |
| Figura 48: Características generales tabla ‘médicos’. | 64 |

| | |
|---|----|
| Figura 49: Características generales tabla ‘user’. | 64 |
| Figura 50: Características generales tabla ‘medición’. | 65 |
| Figura 51: Escenario creado y tabla de estadísticas ‘Sipp’. | 67 |
| Figura 52: Pechera midiendo signos vitales. | 68 |
| Figura 53: Diagrama de flujo para hardware del sistema. | 68 |
| Figura 54: Diagrama de flujo para el software del sistema. | 69 |
| Figura 55: Medición de signos vitales por auscultación dentro del centro veterinario can Kuky. | 71 |
| Figura 56: Medición de signos vitales HOLTER y dispositivo en la pechera. | 71 |
| Figura 57: Medición de signos vitales por auscultación can Nena dentro del centro veterinario. | 72 |
| Figura 58: Medición de signos vitales HOLTER y dispositivo en la pechera. | 73 |
| Figura 59: Medición de signos vitales por auscultación can Ojito dentro del centro veterinario. | 74 |
| Figura 60: Medición de signos vitales HOLTER y dispositivo en la pechera. | 74 |
| Figura 61: Diagrama de barras medición BPMs paciente Kuky. | 76 |
| Figura 62: Diagrama de barras medición SPO2 paciente Kuky. | 77 |
| Figura 63: Diagrama de barras medición BPMs paciente Nena. | 78 |
| Figura 64: Diagrama de barras medición SPO2 paciente Nena. | 79 |
| Figura 65: Diagrama de barras medición BPMs paciente Ojito. | 80 |
| Figura 66: Diagrama de barras medición SPO2 paciente Ojito. | 81 |
| Figura 67: Cálculo de confiabilidad dispositivo HOLTER. | 82 |
| Figura 68: Cálculo de confiabilidad pechera. | 83 |

INDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Ventajas y desventajas del uso de React Native como entorno de desarrollo | 12 |
| Tabla 2. Distribuciones del estándar 802.1x..... | 20 |
| Tabla 3. Frecuencia cardiaca con relación a la edad en perros | 23 |
| Tabla 4. Comparación de características placa Arduino..... | 26 |
| Tabla 5. Comparación de características sensores de pulsioximetría | 29 |
| Tabla 6. Frecuencia cardiaca con relación a la edad en perros | 59 |
| Tabla 7. Mediciones BPMs paciente Kuky. | 75 |
| Tabla 8. Mediciones SPO2 paciente Kuky. | 76 |
| Tabla 9. Mediciones BPMs paciente Nena. | 77 |
| Tabla 10. Mediciones SPO2 paciente Nena..... | 78 |
| Tabla 11. Mediciones BPMs paciente Ojito. | 79 |
| Tabla 12. Mediciones SPO2 paciente Ojito..... | 80 |
| Tabla 13. Tabla de error porcentual referente a las mediciones obtenidas en el día 1. | 85 |
| Tabla 14. Tabla de error porcentual referente a las mediciones obtenidas en el día 2. | 85 |
| Tabla 15. Tabla de error porcentual referente a las mediciones obtenidas en el día 3. | 85 |
| Tabla 16. Tabla de error porcentual referente a las mediciones obtenidas en el día 4. | 86 |
| Tabla 17. Tabla de error porcentual referente a las mediciones obtenidas en el día 5. | 86 |
| Tabla 18. Detalle de precios de hardware utilizado..... | 87 |

RESUMEN EJECUTIVO

En el presente proyecto de investigación se implementa un sistema electrónico de monitoreo de mascotas para la gestión de clínicas veterinarias utilizando VOIP e IOT y así llevar un adecuado manejo y control de los datos obtenidos a partir de un dispositivo medidor de signos vitales y mediante una aplicación desarrollada de forma nativa en React. El sistema electrónico de monitoreo consta de 3 niveles como son adquisición y transmisión de signos vitales, gestión de información y monitoreo y alojamiento en la nube, mismos que se ejecutan de la siguiente manera: para la adquisición y transmisión de datos, la información de signos vitales de la mascota es recolectada mediante un dispositivo electrónico de control diseñado en una pechera que comúnmente los propietarios utilizan para realizar paseos a sus mascotas, esta información es enviada mediante un módulo Bluetooth, para la gestión de información los datos son receptados en el dispositivo móvil mediante la aplicación desarrollada en React Native puro con especificaciones nativas para dispositivos con sistemas Android o iOS, esta aplicación contiene un sistema de inicio de sesión conectada hacia una base de datos en MYSQL instalada en CLOUD de los Servicios Web de Amazon o AWS alojando toda información de la mascota en la nube, también contiene un sistema de gestión para que médicos veterinarios administren sus clínicas de una manera más rápida y funcional, a continuación los datos recibidos son transmitidos al servidor Asterisk alojado en AWS que alerten con el servicio de VOIP al propietario de la mascota sobre alguna anomalía detectada en la medición de datos del mismo.

Este proyecto se basa en contribuir a la gestión de las clínicas veterinarias manejando datos de una manera más sencilla, rápida y eficaz, monitorizando en tiempo real el estado de salud de las mascotas y apoyando a la detección temprana de enfermedades en animales de pequeñas especies.

Palabras clave: Signos vitales, React Native, Bluetooth, aplicación móvil, Asterisk.

ABSTRACT

In the present research project, an electronic pet monitoring system is implemented for the management of veterinary clinics using VOIP and IOT and thus carry out an adequate management and control of the data obtained from a vital signs measuring device and through an application. Natively developed in React. The electronic monitoring system consists of 3 levels such as the acquisition and transmission of vital signs, information management and monitoring and hosting in the cloud, which are executed as follows: for the acquisition and transmission of data, the information of vital signs of the pet is collected through an electronic control device designed in a chest that owners commonly use to walk their pets, this information is sent through a Bluetooth module, for information management the data is received in the mobile device through the application developed in pure React Native with native specifications for devices with Android or iOS systems, this application contains a login system connected to a MYSQL database installed in CLOUD of Amazon Web Services hosting all information of the pet in the cloud, also contains a management system for veterinary doctors manage their clinics in a faster and more functional way, then the data received is transmitted to the Asterisk server hosted on AWS that alerts the pet owner with the VOIP service about any anomaly detected in the measurement of their pet's data.

This project is based on contributing to the management of veterinary clinics by managing data in a simpler, faster and more efficient way, monitoring the health status of pets in real time and supporting the early detection of diseases in small-species animals.

Keywords: Vital Signs, React Native, Bluetooth, Mobile App, Asterisk.

CAPÍTULO I

MARCO TEÓRICO

1.1 Antecedentes Investigativos

En los últimos años se han desarrollado varios trabajos de investigación con relación a los sistemas de telefonía VOIP sobre un CLOUD y teniendo énfasis a resolver problemas de salud que representan una herramienta fundamental para tiempos actuales en donde la movilidad de los propietarios de mascotas se ve afectada por la contaminación habitual del covid-19 configurando un campo de gran interés para la comunidad científica.

Istiadi, Emma Budi Sulistiarini y Rudy Joegijantor, en el año 2015 publicaron el artículo de investigación denominado “Consultation Services Using IVR Telephony Based on Expert System Perspective”, dentro de la Conferencia Internacional sobre ciencia en tecnología de la información (ICSITech), que propone un sistema con perspectiva experta para servicios de consultoría usando la tecnología IVR con diversos métodos que ayudan a canalizar la llamada que se realice, los mecanismos de consulta conducen preguntas y respuestas mediante el uso de respuesta de voz interactiva (IVR) basada en enfoque de sistema experto. Se selecciona un modelo de árbol de decisión que representan la base de conocimientos del sistema. Además, la base de conocimientos está mapeada en un base de datos a la que pueden acceder las aplicaciones. Algunos campos de las tablas de la base de datos contienen una ubicación de los archivos de voz. Esta aplicación reproducirá los archivos de voz de preguntas y recibirá respuestas (código de tono del teclado del teléfono) de los usuarios para cada estado en su camino para llegar a una decisión o conclusión que se le toque al usuario. Este enfoque utiliza el método Fuzzy AHP para el análisis para respaldar la decisión, como resultado de las pruebas indicadas por el seguimiento solicitud y justificación de expertos, el sistema es capaz de rastrear el camino en una

consulta de interacción predeterminada. [1]

En la figura 1 se muestra el diagrama de bloques del sistema experto en línea realizado por los investigadores. El sistema experto es un software que funciona en un sistema informático con una aplicación que proporciona servicio para los usuarios que les permite interactuar mediante una pantalla de bienvenida. Esta interfaz es un medio para expresar los hechos que aparecen de forma sistemática y presentar los resultados de los sistemas de razonamiento.

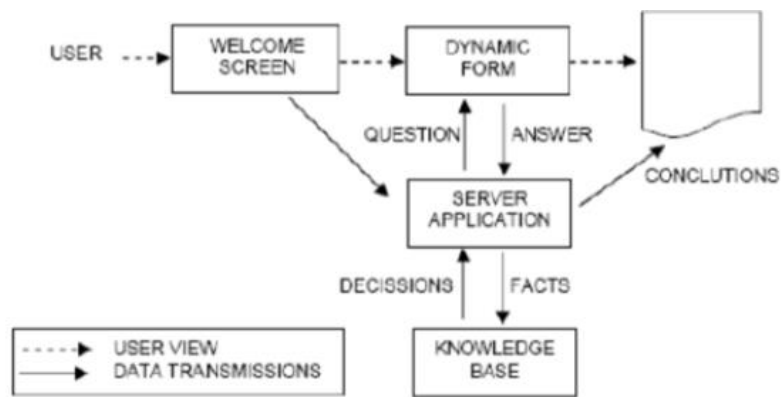


Figura 1: Diagramas de bloque del sistema experto en línea [1]

Archana Gupta, Ajita Verma y Parul Kalra, en el año 2015 publicaron el artículo de investigación denominado “Tele Health Therapy: An Ambient Technology”, dentro de la Conferencia Mundial sobre tecnologías de la comunicación (GCCT 2015), que propone un sistema con ambiente tecnológico mediante el uso de la tele terapia en pacientes que no cuentan con centros médicos cercanos en las localidades que viven. Se genera un sistema de tele salud en la terapia mediante un mecanismo de análisis de medición de pulso, nivel de azúcar, temperatura y otros factores que mediante un analizador de gráficas en tiempo real para así poder diagnosticar de una mejor manera el problema del paciente, la obra presenta un servicio e-Medicare (Tele Health Terapia) que es capaz de proporcionar el tratamiento adecuado en las primeras etapas de las emergencias Médicas. Los investigadores formulan un sistema gráfico de salud

del paciente sobre la base de parámetros recopilados y compararlo con el gráfico ideal formulado sobre la base del formulario de salud llenado por el paciente y con gráfico de alivio inmediato. La terapia intenta administrar el tratamiento inicial con el ayuda de la difusión de la tecnología móvil, base de datos difusa e inteligencia ambiental. [2]

En la figura 2 se muestra el diagrama de bloques del sistema Tele Terapia de Salud por sus siglas en inglés THT diseñado y seguidamente el analizador de resultados. El concepto completo de THT se divide en 4 partes según el trabajo realizado, en primer lugar, se le pide al usuario que complete un registro de sus parámetros de salud y para que las bases de datos THT tengan suficiente conocimiento del historial médico del usuario. A continuación, sobre la base del formulario de registro, las listas de prioridades de los temas de salud están preparados, cuando el paciente siente cualquier problema de salud en el futuro, el dispositivo se centrará en la lista de prioridades de los principales parámetros que el paciente ya padece y da los primeros consejos de salud, asesoramiento y prescripción lo antes posible. Seguidamente con la ayuda del razonamiento y el procesamiento, el dispositivo dibuja un gráfico que muestra la condición real del paciente.

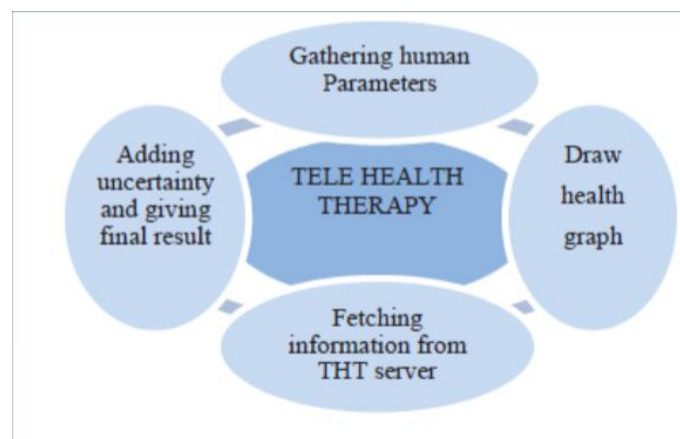


Figura 2: Sistema THT. [2]

En la figura 3 se muestra el trabajo de formación de gráficos de Terapia de Tele-Salud, inicialmente el usuario necesita registrarse llenando su estado de salud en el formulario que se almacenará en el servidor THT para próximo diagnóstico. THT también desarrolla un gráfico de salud sobre la base de los parámetros recopilados desde el formulario de usuario como temperatura, presión sanguínea y nivel de azúcar o glucosa. Cada parámetro está vinculado con su eje respectivo en el área de formación del gráfico.

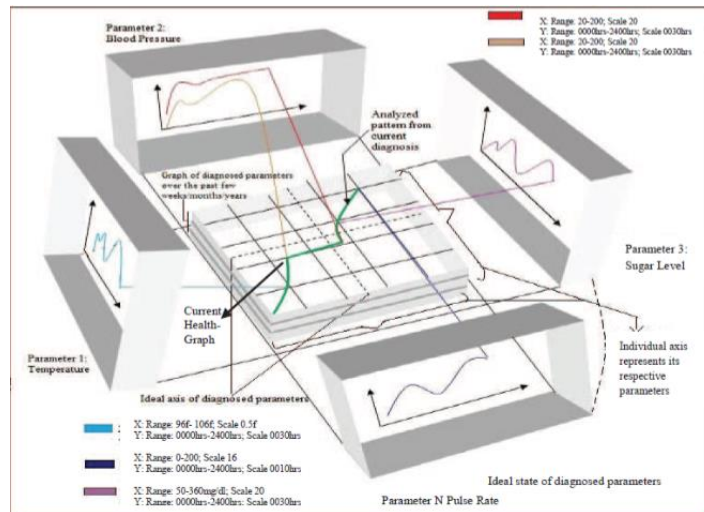


Figura 3: Analizador del sistema THT. [2]

Sarwar Khan y Nouman Sadiq, en el año 2017 publicaron el artículo de investigación denominado “Design and Configuration of VoIP based PBX using Asterisk server and OPNET platform”, dentro del 5to Congreso Internacional de Ingeniería Eléctrica en Pattaya, Tailandia, que propone un sistema VOIP basado en PBX junto con servidor Asterisk dentro de una plataforma OPNET basada en la red y colocada en la nube, con la utilización de un códec de audio y voz con el desarrollo de una red basado en VoIP en todo el campus. Asterisk se configurará con la función de asistente automático, estacionamiento de llamadas, conferencias de llamadas, llamadas en espera, correo de voz, música en espera y notificación por correo electrónico. El proyecto implica el uso de varios softphones y su configuración, el sistema se analizó mediante varias cargas de llamadas para diferentes códecs de voz y los resultados de

diferentes códecs se compararon entre sí para seleccionar el mejor códec para Asterisk PBX. La red se simulará en OPNET (simulador de red) y los resultados que se obtuvieron al valorar las llamadas en situaciones de sobrecargo es que disminuye la calidad de la voz de manera física y con el uso del simulador OPNET. [3]

La figura 4 muestra el diagrama esquemático que se utilizó para el sistema con la utilización del servidor Asterisk. Asterisk es un entorno de código abierto, que se convierte en un puente para el cambio de voz y funciones PBX. Asterisk ayuda a construir una plataforma PBX flexible con hardware económico, que es posible por el host de procesamiento de Multiplexación por división de tiempo TDM y un procesador digital de señales o DSP. Asterisk es una plataforma eficiente que admite múltiples paquetes protocolos de voz, así como también proporciona una aplicación PBX de alto nivel, permitiendo crear una centralita que se compone de las funciones, características y calidad de la telefonía tradicional. En comparación con Asterisk, otras PBX son difíciles de implementar y muy costosas. Este sistema debe implementarse a través de la red y con la ayuda de simuladores se puede encontrar el mejor parámetro que el usuario pueda seleccionar, cuándo se implementará la topología de prueba en un entorno real, por lo que puede causar errores mínimos.

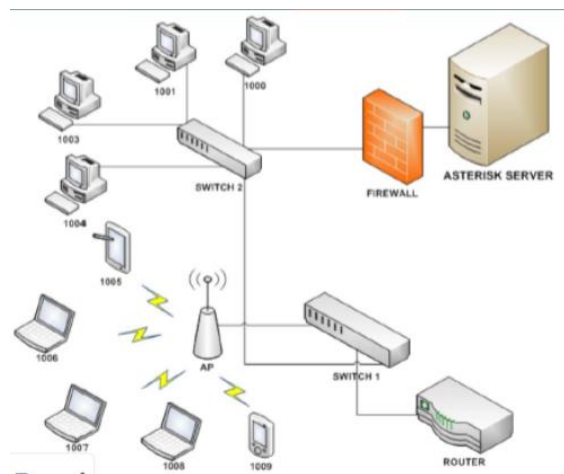


Figura 4: Diagrama Esquemático. [3]

Andrei Tchernykh, Jorge Cortés-Mendoza, Igor Bychkov, Alexander Feoktistov, Loic Didelot, Pascal Bouvry, Gleb Radchenko y Kirill Borodulin, en el año 2018 publicaron el artículo denominado “Configurable cost-quality optimization of cloud-based VoIP”, dentro de la revista Parallel Distributed Computing, que propone un sistema de asignación de estrategias de programación que serán comparadas y ver la mejor para el aprovisionamiento de máquinas virtuales y abaratar costos. La calidad de la voz está influenciada por el estrés o sobrecargo de la CPU, el costo contribuido por la cantidad de horas de facturación para el aprovisionamiento de máquinas virtuales (VM) y las llamadas puestas en espera debido al subprovisionamiento de recursos. Se desarrolla la simulación integral de trabajo real de la empresa MIXvoip junto con estrategias de programación multiobjetivo con horas de facturación, llamadas en espera y criterios de calidad de voz. Introdujeron tres parámetros configurables: Utilización Umbral (UT), umbral de alquiler (RT) e intervalo de predicción (PI) para hacer frente a diferentes preferencias objetivas, cargas de trabajo y propiedades de la nube. Demostraron que los sistemas mejoran el rendimiento, aumentan la calidad de voz y reduce el costo del proveedor de VoIP. En la figura 5 se esquematiza las estrategias de programación multiobjetivo utilizadas para aprovechar los recursos informáticos de los proveedores VOIP y de las máquinas virtuales. [4]

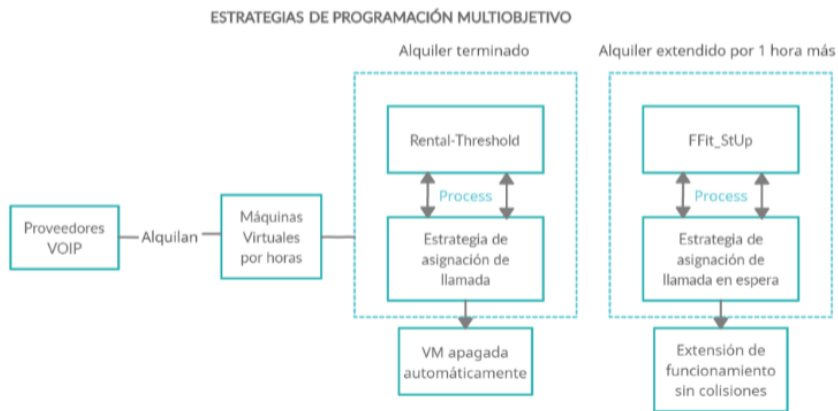


Figura 5: Estrategias multiobjetivo de programación
Elaborado por: El investigador a partir del artículo Configurable cost-quality optimization of cloud-based VoIP [4]

Evan Williams, Zachary Cleghern, Marc Foster, Timothy Holder, David Roberts y Alper Bozkurt, en el año 2020 publicaron el artículo de investigación denominado “A Smart Collar for Assessment of Activity Levels and Environmental Conditions for Guide Dogs” dentro del 42nd Conferencia Internacional Anual de la Sociedad de Ingeniería en Medicina y Biología del IEEE (EMBC), que propone el diseño y la fabricación de un collar inteligente usando el Internet de las cosas, conectado para la recopilación de información ambiental y de comportamiento de los caninos, registrando datos como temperatura ambiente, niveles de ruido ambiental, presión barométrica y humedad relativa, con el uso de sensores que vía Bluetooth se transmiten al teléfono inteligente del propietario, también adquiere la ubicación de la mascota vía GPS y almacenando estos datos en IBM Cloud una vez que el usuario está conectado a una red WiFi. El diseño de bajo consumo del sistema de collar inteligente permite su uso continuo durante 27 horas. Este sistema se está ampliando actualmente para probarse en cientos cachorros caninos por una escuela de perros guía en EE.UU. En la figura 6 se esquematiza el diseño del sistema de collar inteligente que consta de una aplicación móvil que mediante vía Bluetooth registra los datos del collar inteligente que está constituido a base de sensores y a su vez se conecta a una base de datos configurada en un CLOUD de IBM, el dispositivo cuenta con un indicador de batería o carga y es mediante carga inalámbrica. [5]

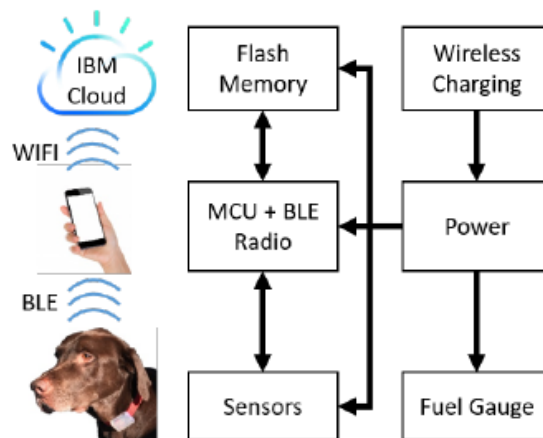


Figura 6: Descripción general del sistema de diseño de collar inteligente.

[5]

Cada uno de estos artículos científicos aportan de diversa manera al proyecto de investigación ya sea en el análisis de los parámetros que se requieren para construir un dispositivo con fines veterinarios es de gran importancia, la forma en que se transmiten, registran y transfieren los datos mediante tecnologías ya estudiadas como Bluetooth o WIFI. Los servidores VOIP más utilizados tanto como sus extensiones y aplicaciones en la interacción humano-máquina. Todo esto es de gran ayuda para que el proyecto de investigación satisfaga las necesidades que tienen las clínicas veterinarias mejorando la gestión y administración de las mismas, enfatizando el cuidado de sus pacientes.

1.2 Contextualización del problema

La presente investigación se basa en la realización de un sistema electrónico de monitoreo de mascotas para la gestión de clínicas veterinarias utilizando VOIP e IoT y poder llevar un control de actividad de pacientes (mascotas) de dichas clínicas veterinarias.

Según un estudio de la Secretaría de Salud del Distrito Metropolitano de Quito, en Ecuador 3 de cada 5 familias tienen una mascota ya sea canino o felino y de acuerdo con estadísticas realizadas a propietarios de mascotas, el 64% acuden a centros veterinarios cuando su mascota se enferma o como manera de prevención, el 19.8% acude a consulta, el 7.8% acude por emergencia, el 22,7% acude a vacunación y el porcentaje restante acude a compras o peluquería canina. [6]

Varias clínicas veterinarias no cuentan con sistemas de monitoreo y control que les permitan conocer información médica de sus pacientes en tiempo real, hoy en día la mayoría de centros especializados en el cuidado de mascotas llevan un registro de sus pacientes ya sean historias clínicas o calendario de vacunación de manera física o en hojas que en momentos determinados pueden provocar la pérdida de esta información relevante.

En Estados Unidos, la proporción de dueños de mascotas con dispositivos electrónicos ya sea de rastreo o de monitoreo de signos vitales por especie sigue aumentando desde 2016, el 43% de los propietarios de caninos, afirmó mediante encuestas realizadas, tener un dispositivo de rastreo electrónico para su mascota. En Ecuador, solo la ciudad de Quito tiene un porcentaje mayor de hogares con gasto mensual planificado para sus mascotas, el cual corresponde al 47%, mientras que en las demás ciudades el propietario de mascotas no considera esta tecnología ya que es muy costosa y por ende tiene una baja demanda. [7]

El servicio de telefonía IP se ubica mayoritariamente en la ciudad de Quito con un 50%, el 43% en Guayaquil, el 8% en Cuenca y el porcentaje restante en otras ciudades. De ellos, el 27%, usan FreePBX (que incluiría a Elastix), el 5% usaría la solución DenwaPBX, el 5% Grandstream, y el 3% Asterisk puro. Las soluciones basadas en Asterisk estarían presentes en un 30% de los dispositivos encontrados. La inversión en telefonía IP en Ecuador pasó de 0 a 16 millones de dólares en 4 años esto es desde el 2009 al 2013 en el sector público, es por esto que la telefonía IP inicia de manera tardía en comparación con otros países más desarrollados tecnológicamente. Las provincias con mayor cantidad de procesos de compra de telefonía IP son Pichincha (56%), Guayas (14%), Azuay (10%) y Tungurahua (6%) de un total de alrededor de 105 procesos realizados durante los últimos 7 años. [8]

En el ámbito medioambiental, la telefonía IP no necesita de un sistema de cableado propio para su funcionamiento por lo que se necesita menos materia prima como cobre o plástico produciendo menor cantidad de residuos. [9]

En cuanto al Internet de las cosas se destaca como una de las soluciones que ofrece una arquitectura de red y una cadena de suministros que impacta y junto a la convergencia de dispositivos como sensores y actuadores que reciben y procesan información digitalizada, para dar solución a necesidades de la vida cotidiana automatizando objetos. [10]

El presente trabajo tiene como finalidad, implementar un sistema electrónico de monitoreo de mascotas para la gestión de clínicas veterinarias utilizando VOIP e IoT basado en CLOUD que permita tener un servicio de consultoría dinámico e interactivo entre usuario y los servicios veterinarios previendo al personal médico todo el historial clínico de sus pacientes. Esto con el fin de aprovechar los recursos tecnológicos para el monitoreo eficaz de mascotas y facilitar la gestión y administración de clínicas veterinarias en cuanto a vacunación e historial médico, buscando obtener mayor número de usuarios en el sector que estén dispuestos a virtualizar este servicio usando tecnologías dinámicas e innovadoras.

1.3 Fundamentación Teórica

1.3.1 Salud en mascotas

La salud en las mascotas se define como el bienestar fisiológico de un animal con ausencia de enfermedades infecciosas y no infecciosas es por ello que un animal que goza de buena salud siempre está contento y alerta. Las ventajas de mantener a una mascota con condiciones ideales de salud es que su compañía se podrá disfrutar de muchos años. [11]

1.3.2 Sanidad móvil

La sanidad móvil permite la monitorización del paciente en su entorno y facilita la recogida de un considerable número de datos médicos personales, así como el acceso de los pacientes a su propia información sanitaria. Sin embargo, el rápido crecimiento de este mercado hace que buena parte de las aplicaciones que se descargan no hayan sido acreditadas por un organismo que garantice su calidad y seguridad. Este desorden digital deja al usuario en una situación vulnerable ante potenciales amenazas de seguridad como virus o ataques al sistema generando lecturas erróneas de valores por ende un diagnóstico equivocado, por lo que ciertas aplicaciones médicas pueden conllevar riesgos para la salud del paciente. Esto hace necesario que se establezcan controles de calidad que permitan que médicos y

pacientes puedan utilizar esta tecnología con completa seguridad para la reducción de posibles errores médicos y la protección de los pacientes. [12]

1.3.3 Entorno de desarrollo

Un entorno de desarrollo es una plataforma en donde el programador de software facilita tareas y fases de programación combinando varias herramientas dentro del análisis, diseño, codificación, pruebas y validación. A continuación, se detallan los diferentes entornos de desarrollo que se analizaron para este proyecto. [13]

1.3.3.1 React

React, también conocida como React.js, es una librería que funciona bajo JavaScript, creada en el año 2013 por desarrolladores de Facebook, siendo de código abierto. Tiene la ventaja de que no sólo se utiliza del lado del cliente sino también puede estar presente del lado del servidor y trabajar en conjunto. Trabaja tomando las actualizaciones del estado de página conforme el desarrollador va creando un nuevo código, ya sea agregando nuevos complementos en la aplicación en desarrollo o agregando lógica en la misma. React se compone de varias funcionalidades como es Virtual DOM o un modelo de objetos de documento creado principalmente para páginas web en donde se muestra el estado actual de la página creada en HTML, también cuenta con componentes puros de JavaScript que transforma el código XML para escribir los componentes compatibles con React.

1.3.3.2 React Native

React Native es un marco de trabajo de JavaScript que crea aplicaciones nativas para iOS y Android y fusiona la creación de aplicaciones móviles nativas en JavaScript con todas las funcionalidades que en esta se aplican como técnicas de gestión de estados, flujo de datos y unidireccional en aplicaciones.

React Native ofrece una experiencia de desarrollo similar al de la Web, como es el caso de la recarga en tiempo real de su aplicación durante el desarrollo. La principal ventaja de React Native es el desarrollo multiplataforma, permitiendo escribir código de funcionalidades similares a través de las plataformas, ya que las interfaces graficas pueden variar según las funcionalidades del sistema en el que se está ejecutando, React Native requiere de una versión Android 6.0 SDK o superior para poder ejecutarlo e instalar todas las dependencias necesarias. En la tabla 1 se puede observar las ventajas y desventajas del uso de React Native como entorno de desarrollo para utilizarlo en este proyecto. [14]

Tabla 1. Ventajas y desventajas del uso de React Native como entorno de desarrollo
Elaborado por: La investigadora [14]

| VENTAJAS | DESVENTAJAS |
|--|---|
| Tiene una curva de aprendizaje relativamente corta en comparación con el aprendizaje que requieren Java. | Poca información a la hora de resolver problemas. |
| Proporciona una mejor experiencia de desarrollador con módulos de terminación rápida como Tabnine. | Si se desea desarrollar aplicaciones para iOS debe de ser en un equipo que corra el sistema operativo del mismo. |
| La depuración móvil es relativamente fácil cuando exista una conexión exitosa al dispositivo móvil, debido a que se ha logrado llevar la depuración web al mundo móvil. Permite crear varias aplicaciones para diferentes plataformas a la vez, ahorrando tiempo en el desarrollo. | Problemas de compatibilidad y depuración: a pesar de ser utilizado por empresas grandes de tecnología, React Native sigue estando es una fase de prueba, lo que en la práctica significa que desarrolladores pueden tener problemas con la compatibilidad de paquetes o las herramientas de depuración. |

1.3.4 Renderizado

Renderizar es la forma que tiene la estructura de programación en cada entorno de desarrollo como aplicación para el caso de React, siempre se renderiza en un elemento del DOM, normalmente un div con un id definido, y se codifica con el método render de la clase ReactDOM, mientras que la renderización en React Native es mediante React props que se refiere a una técnica para compartir código entre componentes en React utilizando una propiedad cuyo valor es una función. En la figura 7 se observa la forma en que se renderiza ReactJS vs React Native con cada una de sus características. [15]

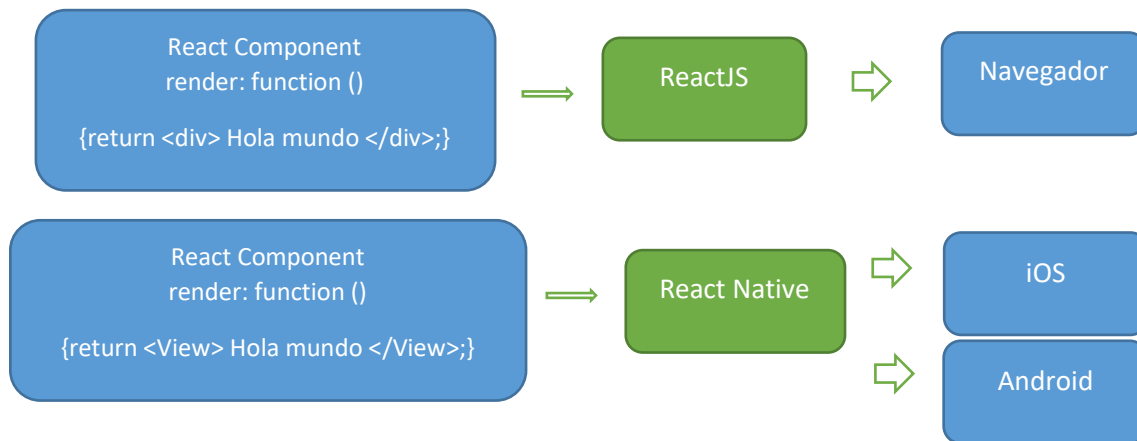


Figura 7: Renderizado ReactJS vs React Native.
Elaborado por: La investigadora [15]

1.3.5 Internet de las cosas

Internet de las cosas (IoT) es la interconexión de dispositivos de detección y actuación que brindan la capacidad de compartir información a través de plataformas con un marco unificado, desarrollando una imagen operativa común para permitir aplicaciones innovadoras [16]

1.3.5.1 COMPONENTES IoT

Los componentes de acuerdo a la arquitectura de un sistema IoT permite que varios objetos separados físicamente estén conectados entre sí mediante una red de comunicaciones, está compuesto por tres elementos básicos: dispositivos o red de sensores, puerta de enlace y nube. En la figura 8 se visualiza los componentes IoT de acuerdo a su arquitectura. [17]



Figura 8: Arquitectura IoT. [17]

En la arquitectura IoT se describen 3 tipos de capas: aplicación, dispositivos y red, en donde cada capa tiene su función específica. La capa de aplicación son todas las áreas en donde IoT es compatible para ser implementado cumpliendo funciones destinadas a seguridad, convivencia, reducción de costos, optimización de procesos, mejorar la calidad de vida de las personas entre otras aplicaciones funcionales.

La capa de dispositivos a su vez es la encargada de especificar los sensores y actuadores que interactúan con el usuario final teniendo en sus lecturas valores reales a los ocurridos en tiempo real, mismos que llegan a la capa de red para

ser almacenados y que se encuentren disponibles a todo momento.

Es por ello que en la capa de red se encuentran todos los dispositivos para establecer y mantener la comunicación entre los dispositivos físicos y las aplicaciones desarrolladas basándose en estándares de comunicación TCP/IP o HTTP para desarrollo de aplicaciones web. Es importante conocer que una etapa fundamental dentro de la arquitectura IoT es el alojamiento en servidores dentro de la nube o cloud los mismos que agrupan las funcionalidades de los servicios informáticos con las aplicaciones conectadas a través del Internet. [18].

Los servicios cloud en general se clasifican según el tipo de servicio y el ámbito en los que se van a desarrollar, según el tipo de servicios se clasifican en infraestructura como servicio con sus siglas en inglés IAAS, este hace referencia a los proveedores que tienen la capacidad de alquilar máquinas virtuales como infraestructura de servicio, siendo un claro ejemplo de ello Amazon web services o AWS, siendo el usuario capaz de rentar servicios virtuales con una gran variedad de CPUs, memoria RAM, almacenamiento y otras características. Otro de los servicios cloud son Software como servicio con sus siglas en inglés SAAS, que se define como la capacidad de alojar software, información ejemplificando a Microsoft Office 365 en donde los usuarios pueden acceder a información en la nube sin necesidad de instalar aplicaciones extras y por último el servicio de cloud como plataforma con sus siglas en inglés PAAS en donde se pueden lanzar aplicaciones como base de datos, servicios de inteligencia y herramientas de desarrollo un ejemplo de esto es Google App Engine enfocado a creación de aplicaciones web online.

La fusión entre sistemas IoT y servicios en la nube ofrece grandes ventajas al momento de desarrollar mecanismos para hacer más fácil la vida diaria de los usuarios, captando información en tiempo real y brindando la posibilidad de

almacenar todos estos valores al instante. De acuerdo con los tipos de procesos que desarrollen cada aplicación existen varios tipos de servidores con los que pueden gestionar la información recabada entre ellos están: Protocolo de Transferencia de Hipertexto (HTTP), Servidor IoT y un servidor de almacenamiento [19]:

- Servidor HTTP.- El servidor HTTP o de Transferencia de Hipertexto se basa en peticiones que el usuario final realiza a un servidor HTTP con datos específicos obteniendo como respuesta datos adicionales e información, es decir para el caso de nuestra aplicación desarrollada en React Native para la gestión de información de pacientes se recurre a la creación de un CRUD en donde se realiza una consulta a la base de datos por medio de peticiones HTTP con los métodos GET y POST y de acuerdo a la función que se debe realizar los datos que retornan se programan en varios métodos y funciones como actualizar, eliminar, editar y más operaciones posibles.
- Servidor de almacenamiento. - El servidor de almacenamiento de datos para aplicaciones IoT es indispensable ya que todos los valores registrados por la capa de dispositivos deben tener un registro de almacenamiento para que siempre estén disponibles es por ello que se utilizó MYSQL alojado en cloud de AWS como gestor de base de datos.

1.3.6 Software de gestión de la nube

El software de gestión en la nube se define como un entorno en que los programadores o desarrolladores controlan y organizan todos sus proyectos ya sea para acceso o inicio de sesión, administrar datos y aplicaciones. Por lo general se implementa un software de gestión en la nube dentro de una infraestructura de servicios en la nube como Servicios Web de Amazon o AWS es por ello que se

detalla a continuación las características del software utilizado dentro de AWS para la gestión de información de nuestra central telefónica en la nube. [20]

AWS es un entorno virtual que permite crear interfaces de servicios web con la funcionalidad de crear y lanzar instancias de diversos sistemas operativos como máquinas virtuales las mismas que cuentan con la capacidad de administrar sus permisos para la red, ejecutar programas sin la necesidad de aplicaciones extras, habilitar puertos TCP y UDP para el sistema de telefonía, entre otras más. Dentro de las características de este gestor de información en la nube se tiene:

- Administración eficiente de datos.
- Infraestructura completamente administrable.
- Cifrado de datos incorporado.

1.3.7 Tecnologías de la información aplicadas en e-Salud

La tecnología de la información aplicadas a la salud digital o e-Salud ha evolucionado de diversas maneras y ha incurrido en diversas áreas de aplicación. Para el campo de la salud se han desarrollado varios sistemas electrónicos donde sus componentes tanto hardware como software ayudan de gran manera al diagnóstico temprano de muchas enfermedades. El hardware analiza y tiene relación directa con el paciente y consta con un microcontrolador que controla cada uno de los procesos mientras que el software cuenta con la interacción virtual con el médico tratante o en el caso de las mascotas con sus propietarios y es el lenguaje de programación el que genera la aplicación ya sean móvil o web.

Para la comunicación de los sistemas electrónicos se manejan de forma alámbrica o inalámbrica y dependen de la utilidad que se les va a dar ya sea para mejorar su portabilidad o para ser más estable. En las comunicaciones

alámbricas se utiliza cables de cobre para la conexión de los dispositivos o un medio guiado como es la fibra óptica, en las comunicaciones inalámbricas a su vez utilizan tecnología sin cables, aunque a diferencia de la comunicación alámbrica que tiene mayor velocidad de transmisión de información esta se ve afectada por la distancia y las pérdidas o retardo de comunicación.

En la clasificación de las comunicaciones inalámbricas se dividen en 3 grupos de acuerdo su aplicación, entre ellos están las WPAN, WLAN y WWAN.

Redes de área personal inalámbrica (WPAN): son destinadas para aplicaciones que facilitan los aspectos de la vida diaria de las personas como aplicaciones en celulares, relojes inteligentes y demás. A continuación, se muestran los diversos estándares compatibles con este tipo de red:

- Bluetooth IEEE 802.15.1.- El estándar IEE 802.15.1 es un estándar de comunicaciones inalámbricas considerado de área personal ya que la distancia en la que su transmisión es aceptable es mínima a comparación de otros estándares, su frecuencia de transmisión está en la banda de los 2.4GHz, es el estándar más conocido ya que es de bajo costo y facilita la administración de aplicaciones ya que se pueden usar en dispositivos portables. El alcance de conexión Bluetooth es de aproximadamente 10 metros, la velocidad de transmisión es de 1-24Mbps y puede trabajar en topología malla, estrella o árbol. [21]
- IrDa (Infrared Data Association). - IrDa es una tecnología basada en rayos luminosos dentro del espectro infrarrojo, permiten comunicación bidireccional, con una velocidad de transmisión de hasta 4 Mbps, opera a una distancia de varias decenas de metros y una comunicación punto a punto. Una de sus ventajas es el bajo costo y consumo de energía, así como también soporta varias plataformas de hardware y software [22].
- Zigbee (IEEE 802.15.4). - El estándar IEEE 802.15.4 que define el nivel

físico y de control de acceso al medio en redes inalámbricas y compite con Bluetooth y Wifi destinado principalmente para aplicaciones de seguridad, monitoreo y control. Se usa mayoritariamente en redes tipo malla ya que permite actualizar de forma dinámica los nodos aportando robustez y rapidez a la red. Cuenta con comunicación bidireccional de bajo costo, a comparación con Wifi, esta tecnología no requiere de un gran consumo de batería y cuenta con mayor alcance [23].

- NFC (Near Field Communication). - NFC se define como la comunicación de campo cercano y es una tecnología de corto alcance de alta frecuencia para el intercambio de información. Trabaja en la banda de los 13,56 MHz, soporta dos modos de funcionamiento, el activo para dispositivos que generan su propio campo electromagnético utilizado para la transmisión de datos y el modo de funcionamiento pasivo que solo un terminal genera este campo y el otro transmite la información. Tiene una velocidad de transmisión de hasta 8Kbps, para su funcionamiento se requiere de una proximidad entre dispositivos de por lo menos 20 centímetros [24].

Para la realización de este proyecto de investigación la tecnología a utilizar es el estándar 802.15.1 o comunicación Bluetooth ya que se considera como una red de área personal con una distancia corta y lo más importante que se puede usar en dispositivos portables como es el caso de la pechera que portará el animal.

Redes de área local inalámbrica (WLAN): Son redes de área local de tipo inalámbricas trabaja con el estándar 802.11 que tienen una geografía limitada, que utiliza la radiofrecuencia y envía una señal de un punto a otro. Presenta ventajas de movilidad, velocidad de transmisión, administración privada de información, bajo costo y escalabilidad, pero a su vez presentan algunas desventajas como dificultad para accesos, inseguridad e interferencias.

El estándar 802.11 establece una estructura por capas para un adecuado funcionamiento del equipo al transmitir y adquirir datos. En tabla 2 se detallan las especificaciones y características de cada una de las distribuciones del estándar 802.11. [25]

Tabla 2. Distribuciones del estándar 802.1x
Elaborado por: La investigadora [25]

| | |
|----------|---|
| 802.11a | Soportando velocidades de hasta 54 Mbps y está regulado en la banda de 5 GHz. Utiliza 52 subportadoras de OFDM. |
| 802.11b | Tiene una velocidad máxima de 11 Mbps y utiliza la banda de frecuencia 2,4 GHz. |
| 802.11g | Velocidad máxima de 54 Mbps y trabaja en la misma banda de frecuencia de 802.11b |
| 802.11i | Dirigido a combatir inseguridades de protocolos de autenticación y codificación abarcando a todos los protocolos 802.1x |
| 802.11n | Velocidad máxima de 600 Mbps y trabaja en la banda de los 2,4 GHz y los 5 GHz. |
| 802.11ac | Taza de transferencia de hasta 433 Mbps. Opera en la banda de 5 GHz. |

Redes de banda ancha móvil (WLAN): Utiliza el estándar 802.11ax son redes de mayor alcance, estas redes utilizan comúnmente los dispositivos móviles ya que tienen la capacidad de conectarse desde cualquier lugar siempre y cuando estén dentro del rango de cobertura. Se dividen en dos áreas tipos, las redes fijas que se utilizan con radioenlace o satélites y las redes móviles que utilizan la compañías o empresas públicas para ofrecer diferentes servicios.

1.3.8 Enfermedades Cardiacas en animales domésticos

Las enfermedades cardiacas en caninos y felinos se definen como cardiopatías se refiere a aquellas enfermedades o problemas anatómicos que afectan al corazón del animal, y que como consecuencia empeoran la calidad de vida e incluso, en casos extremos pueden provocar la muerte del animal, dependiendo de su etiología o de la causa de la cardiopatía, estas pueden ser:

- **Anomalías Congénitas**

Las anomalías congénitas son las más comunes dentro de este tipo, sobre todo las que afectan a los vasos sanguíneos o cámaras cardíacas, como puede ser el conducto arterioso persistente, es decir que, la válvula de salida del ventrículo izquierdo se obstruye dificultando el paso de la sangre, provocando que se hipertrofie el ventrículo para intentar seguir funcionando de manera correcta.

- **Anomalías Adquiridas**

Las anomalías adquiridas pueden aparecer a lo largo de la vida del animal y se deben a alteraciones cardiovasculares, suelen afectar a las válvulas cardiacas o al miocardio. Esta cardiopatía en perros es degenerativa y afecta a las válvulas cardíacas (mitral y tricúspide) y empeora con la edad.

- **Tumores cardiacos**

Los tumores cardíacos normalmente se presentan en perros seniors, entre siete y quince años, el tumor más frecuente es el hemangiosarcoma y su tratamiento suele ser quirúrgico y quimioterapéutico.

- **Arritmia cardiaca**

La arritmia cardiaca es un problema relacionado con el ritmo cardíaco del perro o en su defecto, con el lugar donde se generan los impulsos eléctricos que provocan el latido. Dependiendo del tipo de arritmia que padezca el perro pueden ser más o menos peligrosas.

Los tipos de arritmias son:

Arritmia ventricular.

Arritmia supraventricular.

Problemas en el sistema de conducción.

- **Insuficiencia cardiaca**

Una insuficiencia cardíaca es una alteración del funcionamiento regular del corazón del perro, por la cual no hay una correcta circulación sanguínea, este tipo de cardiopatía afecta a perros de cualquier edad. Provocan síntomas como tos, cansancio, desmayos o una rápida pérdida de peso.

1.3.8.1 Diagnóstico de cardiopatías en perros

Para diagnosticar cardiopatías en perros los médicos veterinarios realizan diversas pruebas para poder determinar un problema cardiaco las cuales son:

- **Auscultación**

En la auscultación se utiliza un estetoscopio con el que se intenta escuchar soplos o ruidos cardíacos extraños, también se tomará el pulso del animal para comprobar el ritmo cardíaco y en la figura 9 se observa el método de auscultación en canes.



Figura 9: Diagnóstico de cardiopatía por auscultación. [26]

- **Electrocardiograma**

Con la toma de un electrocardiograma lo que hacen los veterinarios es registrar la actividad eléctrica del corazón para tener información sobre frecuencia cardíaca, posibles arritmias, etc.

- **Radiografías**

Las radiografías permiten visualizar el corazón, tanto en forma como en tamaño.
[26]

1.3.8.2 Frecuencia cardíaca en perros

La frecuencia cardíaca indica el número de veces en que se contrae el corazón durante un minuto, por este motivo las unidades de la frecuencia cardiaca son latidos sobre minutos, es por ello que en caninos la edad determinará la frecuencia cardíaca de los perros. En esta medición influyen otros elementos externos como el ejercicio que realice o la temperatura del ambiente, los cachorros por lo general tienen valores más altos y estos van descendiendo conforme el animal envejece. En la tabla 3 se visualiza el rango de valores normales de frecuencia cardíaca en perros de acuerdo a su edad.

Tabla 3. Frecuencia cardíaca con relación a la edad en perros [26]

| Edad | Frecuencia cardiaca (Bpm) |
|------------------------|----------------------------------|
| Cachorros (1-12 meses) | 110-120 |
| Adultos (1-9 años) | 90-100 |
| Ancianos (> 10 años) | 70-80 |

El pulso del perro, aunque varíe según el tamaño y la edad, se debería situar siempre entre los 60 y 180 latidos por minuto. Los gatos, sin embargo, tienen frecuencias más altas de entre 140 a 240 pulsaciones por minuto.

1.3.8.3 Saturación de oxígeno en perros (SpO2)

La saturación del oxígeno es un parámetro vital para definir el contenido de oxígeno en la sangre y el lanzamiento del oxígeno, mide el porcentaje de hemoglobina en la sangre y se representa como la saturación arterial del oxígeno (sao2) y saturación venosa del oxígeno (SvO2).

El rango de referencia para la SpO2 es del 90 al 100% en animales que respiran aire ambiental. Un resultado de SpO2 menor puede ser resultado de una hipoxia moderada o paro cardiaco. [27]

1.3.9 Adquisición y monitoreo de los signos vitales

Para la adquisición y monitoreo de signos vitales existen varios sistemas para la medición y el control de las diferentes constantes vitales en caninos que por medio de médicos veterinarios son analizados y medidos. Los sistemas se encargan de adquirir los datos de manera eficaz y precisa, a partir de este punto se monitorean para el análisis del estado del paciente.

Para la medición del pulso es necesario colocar un monitor que con pulseras de acuerdo al grosor de la pata del animal se determinan estos valores ya sean frecuencia cardiaca, saturación de oxígeno en la sangre y pulsos.

En el mercado se puede encontrar una gran variedad de tecnologías para el monitoreo de los signos vitales para animales. Existen desde dispositivos dedicados y portátiles como el caso de pulsímetros, lectores de onda electrocardiogramas, sensores de temperatura y más.

Los dispositivos similares a este proyecto son:

- **PetPace.**- PetPace es un collar inteligente que monitorea la salud y la actividad de tu perro y recibe alertas de salud a tu Smartphone. Monitorea variables de salud como temperatura, pulso, respiración, variabilidad de frecuencia cardíaca (HRV), nivel de actividad, postura, calorías. Todos los datos se graban y procesan en el collar PetPace y se transmiten de forma inalámbrica a una base de datos segura en la nube. Las alertas se envían inmediatamente si hay una causa de preocupación. Los datos se actualizan continuamente las aplicaciones móviles o web, tiene un costo en Europa de 20 euros. [28]
- **Memopet.** - Memopet es un collar inteligente distribuido a nivel nacional con tecnología NFC incorporada que, a través de la aplicación gratuita memopetID, permite insertar en el collar toda la información de salud de su mascota que siempre viajará con él. Los datos siempre estarán disponibles y visibles en los Smartphone a través de la aplicación o el lector NFC. El collar se convierte en una base de datos y cédula de identidad digital. Toda la información personal y de salud de su mascota viajará con él, mientras que los datos siempre estarán disponibles en los Smartphone a través de la aplicación para teléfonos inteligentes memopetID, basta acercar el teléfono al broche del collar para acceder a todos sus datos. El collar Memopet tiene un costo de \$28 dentro del territorio Nacional. [29]

1.3.10 Placa de Desarrollo

La placa de desarrollo o placa única por sus siglas en inglés SBC con un procesador, memoria, unidades de entrada y salida y características propias de funcionalidad. A continuación, se detalla la placa de desarrollo Arduino con cada una de sus especificaciones:

- **Arduino.** - Arduino es una plataforma de código abierto basada en hardware y software libre, flexible para desarrolladores, consta de un microcontrolador ATMEL en la que se pueden grabar y programar instrucciones para diversas funcionalidades. El microcontrolador de Arduino posee una interfaz de

entrada donde se pueden conectar diversos periféricos, también posee una interfaz de salida encargada de llevar información desde el Arduino hacia otros periféricos. Existen diversas placas de Arduino con diversas características, en la tabla 4 se realiza un análisis de cada una de ellas y así poder elegir la más propicia para el momento de su implementación. [30]

Tabla 4. Comparación de características placa Arduino [30]

Elaborado por: La investigadora

| | Arduino Uno | Arduino Leonardo | Arduino Mega | Arduino Nano | Arduino Pro | Arduino Pro Mini |
|------------------------------------|--------------------|-------------------------|---------------------|---------------------|--------------------|-------------------------|
| Micro controlador | ATmega328 | ATmega32u4 | ATmega2560 | ATmega168 | ATmega168 | ATmega328 |
| Voltaje | 5 V | 5V | 5V | 5V | 5V | 3.3 V |
| Pines Digitales | 14 | 20 | 54 | 14 | 14 | 14 |
| Pines Digitales PWM | 6 | 7 | 6 | 6 | 6 | 6 |
| Pines Analógicos de entrada | 6 | 12 | 6 | 8 | 6 | 6 |
| Corriente DC | 50 mA | 50 mA | 50 mA | 40 mA | 40 mA | 40 mA |
| Memoria Flash | 32 KB | 32 KB | 256KB | 16 KB | 16 KB | 32 KB |
| SRAM | 2 KB | 2.5 KB | 8 KB | 1 KB | 2 KB | 2 KB |
| Velocidad | 16 MHz | 16 MHz | 16 MHz | 16 MHz | 16 MHz | 8 MHz |
| Longitud | 68.6 mm | 68.6 mm | 101.52 mm | 43.2 mm | 40 mm | 33 mm |
| Ancho | 53.4 mm | 53.3 mm | 53.3 mm | 18.5 mm | 18 mm | 18 mm |
| Peso | 25 g | 20 g | 37 g | 7 g | 5 g | 2 g |

Analizando todas las características técnicas y funcionales de cada una de las placas Arduino de la tabla 4, se determinó que la placa Arduino Pro mini es la indicada para utilizarla en este proyecto ya que este sistema de monitoreo será ensamblado en una pechera que lo usan animales de compañía y para que esta sea portable se requiere que las dimensiones del dispositivo sean muy pequeñas, es por esto que la placa Arduino Pro mini es la placa más adecuada y con todas las características que se requieren al momento de conectar con los sensores y pantalla oled.

1.3.11 Sensores de pulsioximetría

Un sensor de pulsioximetría es un dispositivo que mide los niveles de oxígeno en la sangre y por ende el ritmo cardíaco cuando el corazón late. La selección de este sensor depende de la aplicación a la que este destinado para este caso hacia el montaje en una pechera que utilizará una mascota, la precisión y exactitud de este sensor depende de su marca y de acuerdo a estudios realizados por la Sociedad Americana de Hematología, diferentes modelos de oxímetros de pulso poseen un margen de error de un +/-2% a un +/-10%. [31]

Esta investigación implica indagar sobre el sensor más adecuado que cumpla con los requerimientos de bajo costo, fácil implementación y sin complicaciones, que entre los más relevantes se encuentran:

Sensor MAX30100 sensor de latido del corazón y oxímetro de pulso

El sensor MAX30100 es un circuito integrado de montaje que contiene sensores para medición de ritmo cardíaco y saturación de oxígeno en la sangre, se aplica como un pulsioxímetro y sensor de ritmo cardíaco en monitores médicos y dispositivos para entrenamiento en deportistas, el dispositivo consta de un led rojo, un infrarrojo y un fotodetector, el led rojo se encarga de la oximetría de pulso, el led infrarrojo mide el ritmo cardíaco, cada led emite luz al colocarse en la piel en donde el fotodetector detecta las variaciones en la luz por cambios de volumen en la sangre. En la figura 10

se puede visualizar el circuito integrado que funciona como sensor de pulsioximetría. [32]



Figura 10: Sensor MAX30100. [32]

Sensor de oximetría de pulso NJL5501R

El sensor NJL5501R es un circuito integrado de montaje que contiene está compuesto de un LED rojo e infrarrojo y un fotodetector de alta capacidad, sensor óptico adecuado para lecturas de frecuencia cardíaca y oximetría de pulso. En la figura 11 se visualiza el sensor NJL5501R en su forma física. [33]

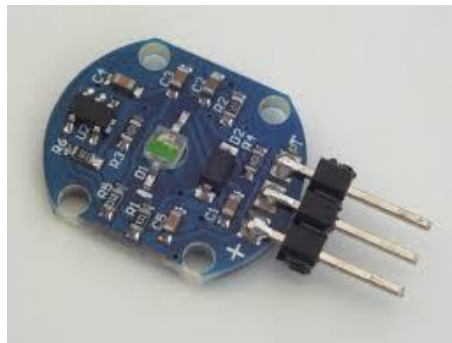


Figura 11: Sensor NJL5501R. [33]

Sensor de oximetría de pulso y frecuencia cardíaca SFH7050

El sensor SFH7050 es una placa compuesta de 2 funcionalidades, un pulsioxímetro SFH7050 y una maximización análoga frontal TI AFE4404 para la biodetección y monitorización de variables vitales, consta de 3 LED de emisión de luz roja, verde e infrarroja en contacto con la piel o el torrente sanguíneo, también tiene un fotodetector para detectar la absorción de luz en la sangre. En la figura 12 se visualiza el sensor SFH7050. [33]



Figura 12: Sensor SFH7050. [33]

A continuación, se detalla la tabla 5 como comparación de los sensores antes mencionados y seleccionar el adecuado para este proyecto de investigación.

Tabla 5. Comparación de características sensores de pulsioximetría [33]

Elaborado por: La investigadora

| | Sensor MAX30100 | Sensor NJL5501R | Sensor SFH7050 |
|--|------------------------------------|--|---|
| Longitud de onda de emisión | Rojo (650 nm) IR(950 nm) | Rojo (660 nm \pm 3 nm) IR(940 nm \pm 10 nm) | Rojo (640 nm \pm 5nm) IR(940 nm \pm 5nm) |
| Interfaz digital | I2C | I2C | I2C |
| Voltaje | 1.8 – 3.3 V | 3.3 – 5V | 3.3 – 5V |
| Precio | \$9 | \$18 | \$40 |
| Compatibilidad con Arduino | SI | SI | NO |
| Luz Ambiental | Sistema de cancelación luz externa | NO | Sistema análogo frontal |
| Sistema de movimiento | Resistencia al movimiento robusto | NO | NO |
| Precisión aproximada | 97% | 89% | 93% |
| Dispositivo emisor de radiación | NO | NO | NO |

De acuerdo a la tabla 5, el sensor MAX30100 es el indicado para la realización de este proyecto ya que todas sus características técnicas y de funcionalidad cumplen con los requerimientos del proyecto.

1.3.12 Pantalla OLED

La pantalla OLED es un display que se destaca por su gran contraste y mínimo consumo de energía debido a que pixel genera su luz y no requiere retroiluminación como los LCD la pantalla OLED 0.91" posee una resolución de 128*32 pixeles, tiene una interfaz de comunicación I2C diseñado para trabajar de 3.3V a 5V sin necesidad de convertidores, consta de pines para conexión a placa Arduino como son SDA o para sistema de datos y SCL para sistema de reloj. En la figura 13 se observa la pantalla OLED con la ubicación de sus pines de conexión. [34]



Figura 13: Pantalla OLED 0.91". [34]

1.4 Objetivos

1.4.1 Objetivo General

Implementar un sistema electrónico de monitoreo de mascotas para la gestión de clínicas veterinarias utilizando VOIP e IoT.

1.4.2 Objetivos Específicos

- Analizar el funcionamiento de los sistemas de monitoreo de mascotas y la gestión de asistencia técnica en clínicas veterinarias.
- Implementar un collar electrónico inteligente utilizando Internet de las cosas.
- Implementar el sistema de gestión de asistencia técnica en clínicas veterinarias utilizando VOIP.

CAPÍTULO II

METODOLOGÍA

2.1. Materiales

Para la elaboración de este proyecto de investigación se utilizaron varios materiales tanto como hardware y software con el fin de diseñar e implementar un sistema electrónico de monitoreo de mascotas, en la parte de hardware se diseñó un circuito electrónico con un microcontrolador Arduino Pro mini, sensor MAX30100, pantalla OLED, módulo de comunicación inalámbrica Bluetooth, batería de polímetro de litio, para el software se emplearon herramientas como el entorno de programación React Native, instancia centos 7 de Amazon, central Asterisk, aplicación Zoiper para emular las llamadas IP, teléfono Smartphone, fuentes de investigación como tesis, libros y artículos. Además de contar con información y registros de pacientes de la Clínica Veterinaria Mascotas Felices perteneciente al cantón La Maná.

2.1.1. Encuestas

Para la selección de Software de este proyecto se requiere realizar una previa investigación en los centros veterinarios de la localidad del Cantón La Maná Provincia de Cotopaxi en el cual se plantea varias interrogantes para conocer si es necesario la creación de un sistema web para el monitoreo de datos de las mascotas para el personal médico o si la generación y desarrollo de una aplicación móvil es una herramienta más óptima, fácil y portable.

Esta encuesta fue realizada de manera virtual con la ayuda de un generador de encuestas llamado questionpro.com hacia 4 médicos veterinarios que tienen o están a cargo de clínicas veterinarias en la ciudad de La Maná. En la figura 14 se observa los nombres de los médicos veterinarios encuestados.

| Escriba su nombre y que título universitario posee | | |
|--|----------|--|
| 11/30/2021 | 60148099 | Genesis Barros médico veterinario |
| 11/25/2021 | 59821901 | Jefferson León médico veterinario y zootecnista |
| 11/25/2021 | 59796334 | Marcelo Rivadeneira-médico veterinario y zootecnista |
| 11/24/2021 | 59756448 | Mario Santiago Sarzosa troya . Médico veterinario |

Figura 14: Médicos veterinarios encuestados.

Fuente: Questionpro.com

Las preguntas realizadas fueron las siguientes:

1. ¿Cómo registra el historial médico de sus pacientes?

De acuerdo con las respuestas de los médicos veterinarios 3 de ellos registra la información e historial médico de sus pacientes de forma física mediante carnets o fichas, uno de los médicos registra la información de manera virtual en documentos de Word dentro de su ordenador, pero no del total de sus clientes ya que algunos de sus registros también los tiene de manera física en carnets de vacunación y hojas de historial médico. En la figura 15 se observa el diagrama de barras de los resultados obtenidos en la pregunta 1.

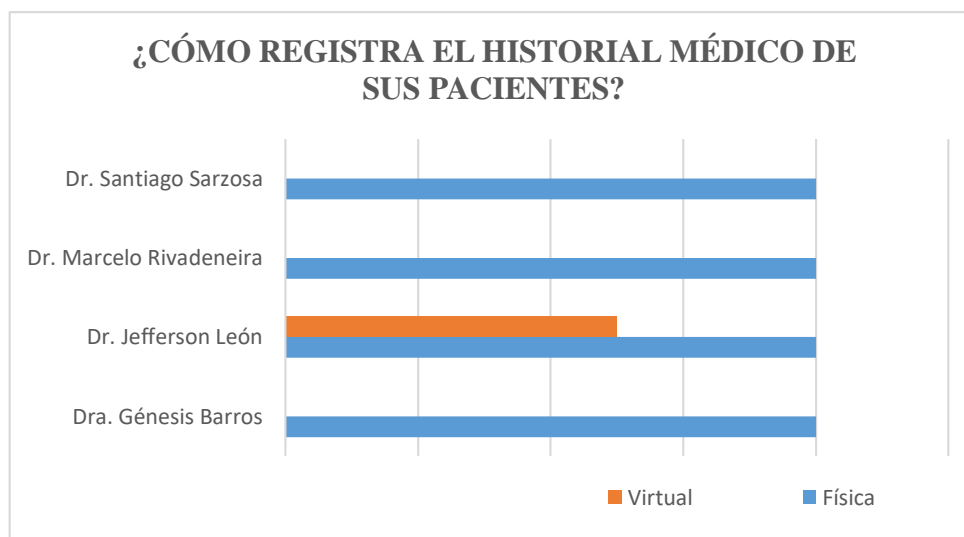


Figura 15: Diagrama de barras resultados primera pregunta.

Fuente: Questionpro.com

2. ¿Con qué frecuencia pierden los carnets de vacunación sus clientes?

La frecuencia de pérdida de los carnets de vacunación y registro abarca al 50% de médicos encuestados a que a veces los propietarios de sus pacientes pierden dicha información, otro médico manifestó que casi siempre lo pierden y el último manifiesta que en su caso casi siempre es la pérdida de esta cartilla. En la figura 16 se visualiza el diagrama de barras de los resultados de la segunda pregunta.

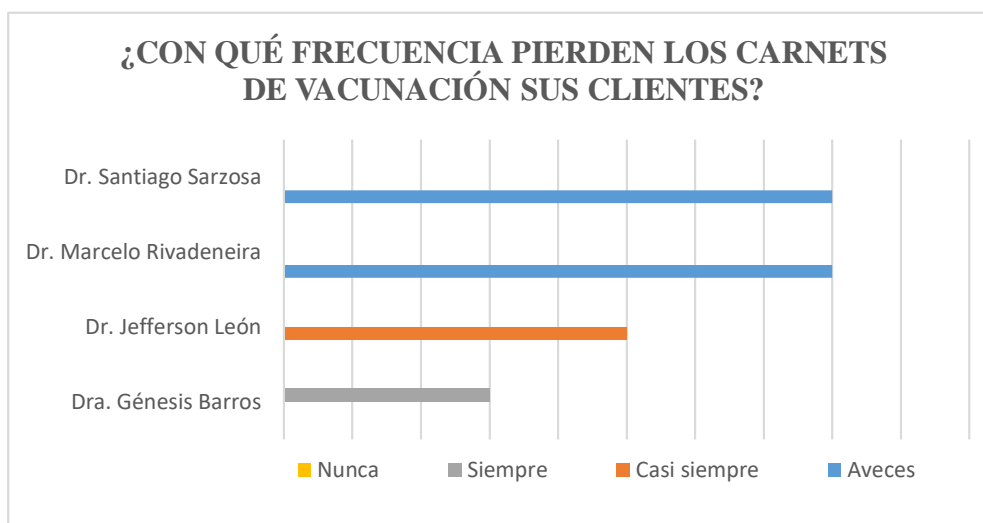


Figura 16: Diagrama de barras resultados segunda pregunta.

Fuente: Questionpro.com

3. ¿Accedería usted a virtualizar la información de sus pacientes?

Para este caso el total de médicos encuestados, el 100%, están dispuestos a virtualizar o digitalizar la información de sus pacientes, ya que piensan que es una forma más segura y ágil de utilizar esta información. En la figura 17 se observa el diagrama de barras del total de encuestados afirmando esta pregunta.

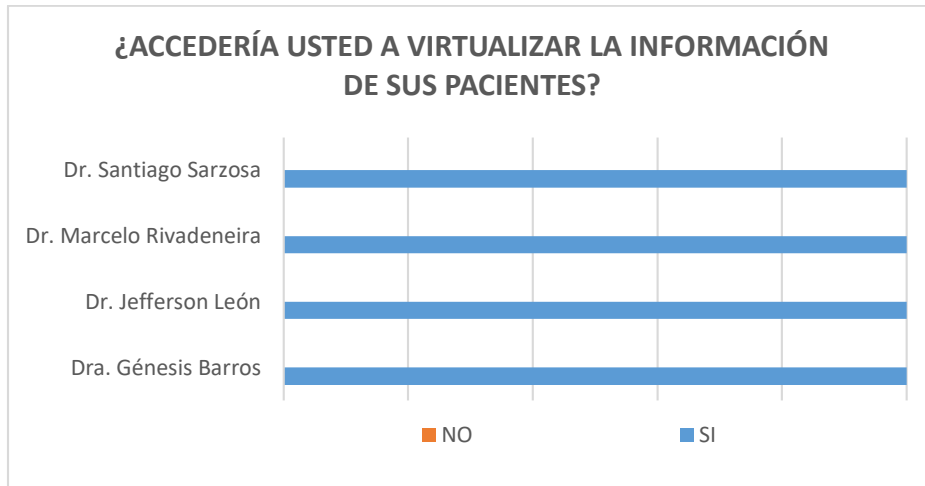


Figura 17: Diagrama de barras resultados tercera pregunta.

Fuente: Questionpro.com

4. Si su respuesta anterior fue Si, ¿de qué manera le facilitaría su trabajo al monitorear los datos de sus pacientes?

En esta última pregunta de igual manera el 100% de encuestados cree que la mejor opción de manejo de una plataforma virtual para administrar su información es por medio de una aplicación móvil. En la figura 18 se observa el diagrama de barras de los resultados de esta pregunta con su aceptación hacia la creación de una aplicación móvil.

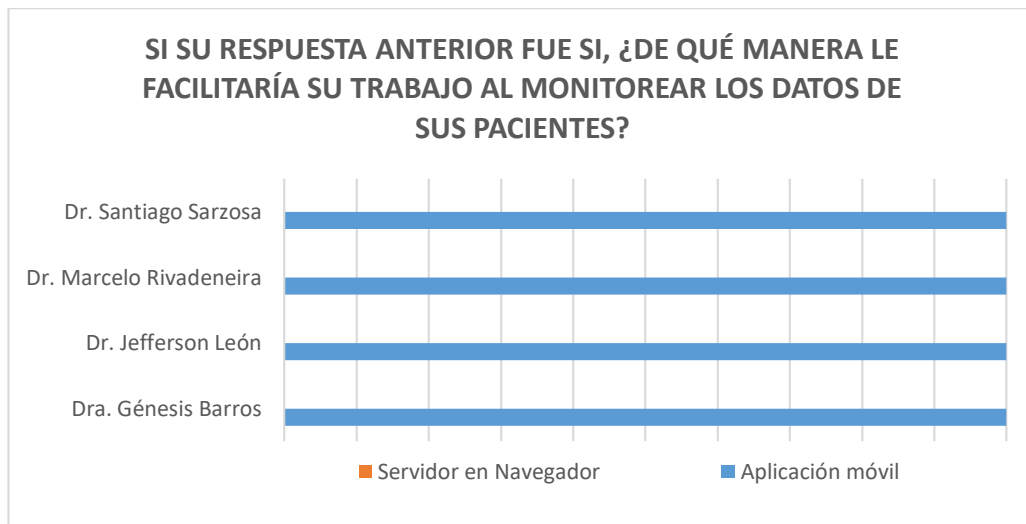


Figura 18: Diagrama de barras resultados cuarta pregunta.

Fuente: Questionpro.com

Resultados Generales de la Encuesta

Se realizó una encuesta virtual a 4 médicos veterinarios que tienen sus negocios en el cantón La Maná Provincia de Cotopaxi mediante una herramienta virtual llamada Question.pro que ayuda a obtener las estadísticas de las respuestas de una encuesta determinada y mediante esta encuesta se obtuvo como resultado que actualmente la mayoría de ellos administran la información de sus pacientes como registro de vacunas e historial médico mediante cartillas o carnets de manera física, solo uno de ellos lo hace de manera virtual en documentos de Word en plantillas diseñadas, asegurando que esta información manejada de esta manera es vulnerable a pérdidas, también se demostró mediante esta encuesta que todos ellos están dispuestos a actualizar su información mediante medios tecnológicos que ayudarían de gran manera a la gestión y administración adecuada de sus negocios, así como la agilidad para detectar enfermedades en sus pacientes y llevar un registro de todas las vacunas suministradas a los mismos por medio de una aplicación móvil en su teléfono celular.

2.2 Métodos

2.2.1 Modalidad de investigación

Investigación Aplicada

Para el desarrollo del presente proyecto se utilizó investigación aplicada, debido a que se pondrá en práctica los conocimientos adquiridos para la implementación y programación de un sistema electrónico de monitoreo, así como de una centralita telefónica virtualizada utilizando códigos de programación en el servidor instalado en CLOUD para la gestión y administración de clínicas veterinarias.

Investigación Bibliográfica

Para el desarrollo del trabajo de titulación fue necesario una revisión bibliográfica en libros, revistas, artículos y bases de datos científicas de diversos repositorios de las Universidades en Ecuador e información web para conocer los requerimientos y especificaciones de los sistemas electrónicos, teleasistencia, telefonía VOIP sobre Internet e IoT; siendo de gran ayuda.

Investigación Experimental

La presente investigación es de carácter experimental, ya que a través de pruebas mediante emuladores se analizan las ventajas que tiene el sistema electrónico de monitoreo y para el sistema de consultoría virtual para clientes en el ámbito de la medicina especialmente para propietarios de animales catalogados como pequeñas especies (canino y felino).

2.2.2 Recolección de información

Para la recolección de la información de diseño se analizaron artículos, libros, estudios y tesis desarrolladas en los últimos años, cada uno de estos relacionados a sistemas de teleasistencia, a sistemas de telefonía VOIP mediante el uso del protocolo de Internet y a dispositivos realizados y relacionados al monitoreo de mascotas. Además, se recopilará información médica de varios pacientes para la creación de base de datos y su consulta en la ciudad de La Maná.

2.2.3 Procesamiento y análisis de datos

Para el procesamiento y análisis de datos se llevará a cabo los pasos descritos a continuación:

- Lectura comprensiva de la información recopilada.
- Análisis de la información recabada haciendo énfasis en los puntos más

importantes para el proyecto de investigación.

- Recopilación de datos técnicos de los elementos a utilizar mediante la utilización de fuentes bibliográficas de alto nivel, tesis, artículos científicos, etc.
- Interpretación y análisis de sintomatología en canes con problemas cardiacos.
- Interpretación de información obtenida.
- Planteamiento de propuesta de solución.
- Presentación de resultados de acuerdo con los objetivos planteados.

2.2.4 Propuesta de Solución

La implementación del sistema electrónico de monitoreo de mascotas para la gestión de clínicas veterinarias utilizando VOIP e IOT junto con CLOUD, permitirá aprovechar de manera eficiente los recursos tecnológicos para servicios de consultoría médica entre usuario y profesional veterinario, virtualizando y digitalizando datos e información de pacientes en clínicas veterinarias que actualmente se archivan de manera física identificando a la brevedad problemas médicos principales con la ayuda del collar inteligente de medición de signos vitales. El sistema constará de un menú interactivo y dinámico con el usuario para poder monitorear los parámetros de la mascota, los datos se envían al CLOUD donde se procesarán y analizarán a través de un servidor, dicho servidor consta de una centralita telefónica en donde el usuario puede hacer consultas y en el caso de existir una variable fuera del rango normal de actividad alertar inmediatamente al propietario de la mascota para que acuda al centro veterinario como una emergencia médica.

2.2.5 Desarrollo del Proyecto

Para cumplir con los objetivos planteados en el proyecto de investigación se llevó a cabo las siguientes actividades:

1. Análisis, descripción y evaluación de los sistemas de monitoreo de mascotas que existen actualmente en el mercado.
2. Análisis y descripción de la gestión actual de asistencia técnica en clínicas veterinarias.
3. Diseño del collar inteligente de medición de signos vitales.
4. Lectura y registro de los datos recopilados por los sensores.
5. Identificación y selección del CLOUD, servidor y sistema operativo indicado.
6. Envío y recepción de datos al CLOUD.
7. Implementación del algoritmo adecuado para el control del sistema VOIP.
8. Simulación del funcionamiento de un sistema de teleasistencia médica veterinaria usando emuladores de voz.
9. Validación del algoritmo de control del sistema de teleasistencia.
10. Establecer una conexión virtual de una llamada entre dos puntos.
11. Realización de pruebas de funcionamiento del sistema.
12. Validar el sistema de gestión de asistencia técnica en una clínica veterinaria utilizando VOIP.
13. Elaboración del informe final de la propuesta proyecto.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de los resultados

La implementación de un sistema de monitoreo de mascotas para la gestión de clínicas veterinarias utilizando VOIP e IoT, ayuda de cierta manera a la administración de las clínicas veterinarias gestionando la información de sus pacientes de correcta manera y estará siempre disponible en los teléfonos móviles tanto de los propietarios de mascotas como de los médicos de turno.

3.2 Análisis de Factibilidad

Factibilidad Técnica

El presente proyecto es factible en el ámbito técnico ya que todos los dispositivos como pantalla, sensores, placa Arduino, módulo Bluetooth han sido estudiados y son comerciales, por lo cual pueden ser adquiridos dentro del país y a un precio accesible.

Factibilidad Económica

El proyecto es económicamente factible porque todo gasto que implica en la realización del proyecto son a cargo de la investigadora y tomando en cuenta que son elementos o dispositivos que se encuentran fácilmente en tiendas de elementos electrónicos y en el caso de pagos de servicios en la nube también se solventaron mediante pagos virtuales.

Factibilidad Bibliográfica

La información requerida para el proyecto es de libre acceso, la cual se obtuvo mediante libros, publicaciones científicas, tesis referenciales, conocimiento adquirido entre otros.

3.3 Desarrollo de la propuesta

El sistema electrónico de monitoreo de mascotas está compuesto de tres etapas: el sistema de adquisición y transmisión de signos vitales, la gestión de información y monitoreo de datos mediante la aplicación móvil desarrollado en React Native, así como el registro de vacunación con un control de logeo para médicos y clientes, por último, el alojamiento de información en la nube y un asistente virtual con VOIP mediante Asterisk instalado en CLOUD de Amazon Web Services. En la figura 19 se observa el esquema general del prototipo.

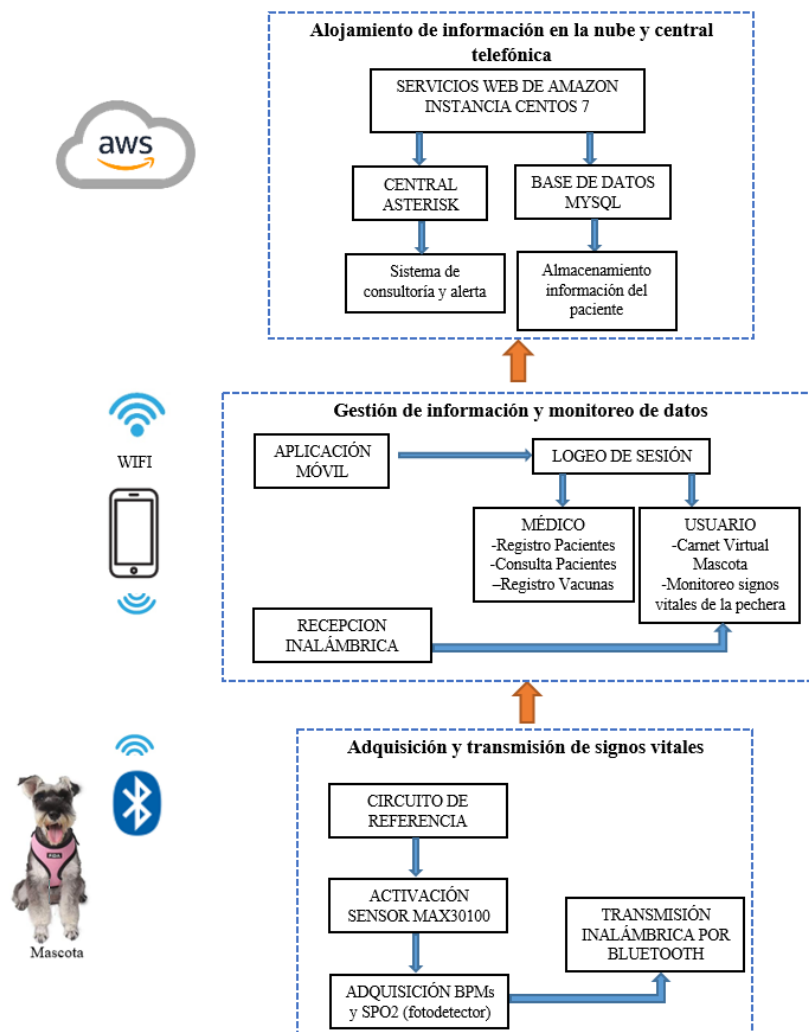


Figura 19: Estructura general del sistema de medición de signos vitales

Fuente: Elaborado por la investigadora

3.3.1 Adquisición y transmisión de signos vitales

Dentro del sistema de adquisición de datos, la portabilidad es una de las características importantes para que el sistema cumpla con todos los parámetros establecidos, por este motivo se diseñó el dispositivo de adquisición de datos en forma de un collar en conjunto con una pechera. El dispositivo es liviano y portable por la mascota con los sensores ubicados correctamente para la adquisición de datos.

El proyecto de sistema electrónico de monitoreo de mascotas para la gestión de clínicas veterinarias utilizando VOIP e IoT comienza con el diseño y construcción de un dispositivo medidor de signos vitales adecuado en una pechera que adquiere los valores de ritmo cardiaco y saturación de oxígeno en la sangre. En la figura 20 se muestra un diagrama del funcionamiento de la pechera para adquisición y transmisión de datos.

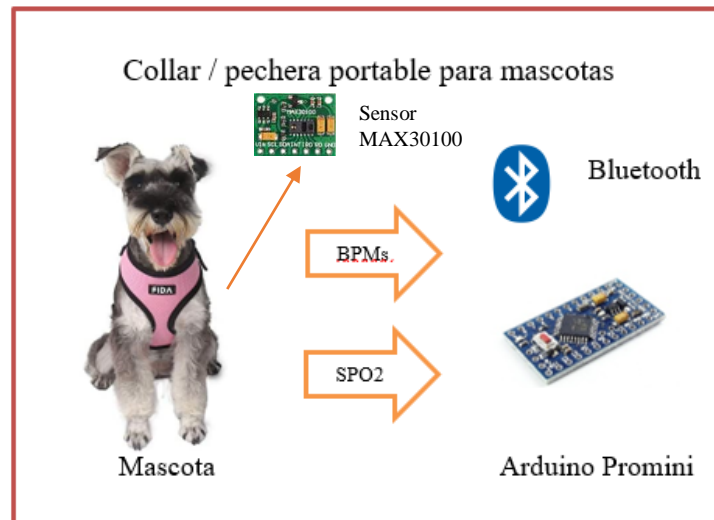


Figura 20: Estructura del sistema de adquisición y transmisión de datos

Fuente: Elaborado por la investigadora

La pechera se compone de tres partes: el procesamiento de los datos, la capa de sensado y el sistema de energía. En el procesamiento de datos se utilizó la placa de desarrollo Arduino Promini 3.3 basada en el procesador ATmega328 funcionando a 8MHz de oscilación, ya que cuenta con el protocolo de comunicación inalámbrica Bluetooth que se utilizan para la adquisición de datos junto con el módulo HC-05. En el Anexo A se encuentra detallado la distribución de los pines con las que cuenta esta placa de desarrollo y sus características más importantes.

El programa en el que se desarrolló el software para el sistema electrónico es Arduino IDE, que utiliza un lenguaje propio de programación desarrollado para las placas Arduino, además, es de código abierto para realizar programación en varios microcontroladores, se inició indagando el funcionamiento, la conexión de los pines y las librerías necesarias para el sensor pulsioxímetro MAX30100 de tal manera que se comunique con la placa de Arduino Pro mini programado en el software Arduino IDE. Inicialmente se instala e incluye la librería del sensor MAX30100, pantalla OLED de las dimensiones exactas del dispositivo físico y la librería correspondiente para comunicación I2C estas están disponibles en el gestor de librerías para Arduino IDE y se incluyen de la siguiente manera. En la figura 21 se observa el código de las librerías incluidas para reconocimiento de la comunicación I2C, de la pantalla OLED y sensor MAX30100.

```
#include <Wire.h>//Librería para comunicación I2C
#include "MAX30100_PulseOximeter.h"//Librería para reconocimiento de sensor MAX30100 como extension pulsioxímetro
#include "MAX30100.h"//Librería para reconocimiento de sensor MAX30100
#include <U0g2lib.h>//Librería para habilitar pantalla OLED
```

Figura 21: Código de librerías utilizadas en la programación Arduino IDE.

Fuente: Elaborado por la investigadora

Para la capa de sensado se utilizó el sensor MAX30100 con comunicación I2C, este sensor cuenta con un pulsioxímetro, que trabaja con dos emisores y un receptor de luz para captar la cantidad de luz reflejada por el paciente y así calcular la saturación

de oxígeno en sangre y el ritmo cardiaco del paciente, se eligió este sensor ya que cuenta con las dimensiones adecuadas para ser utilizado como un collar en animales de pequeña especie.

Acondicionamiento del sensor MAX30100

El sensor MAX30100 trabaja con tecnología óptica, un suministro de energía de 1.8 a 3.3 voltios y comunicación serial I2C. Este sensor adquiere datos como ritmo cardiaco y saturación de oxígeno en la sangre mediante un led rojo y uno infrarrojo. La emisión de luz roja del diodo interno del sensor varía de acuerdo al ritmo con que el corazón late, esta luz se refleja en las venas o arterias y son receptadas por el fotodiodo. En la figura 22 se observa un diagrama explicativo del principio de funcionamiento del sensor MAX30100.

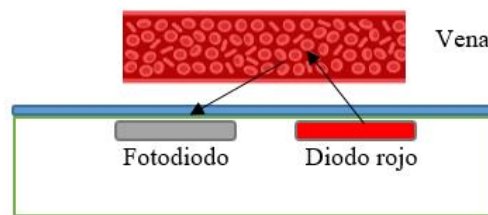


Figura 22: Principio de funcionamiento sensor MAX30100.

Fuente: Elaborado por la investigadora

El sensor capta los valores de frecuencia cardiaca (HR) y saturación de oxígeno (SPO2), corrige estas señales mediante la desviación generada por la luz ambiental, temperatura y ruido eléctrico hasta convertirlas en un valor ADC de conversión analógica-digital almacenandolas en un buffer para poder ser utilizadas en un microcontrolador.

Para acondicionar este sensor se analiza las longitudes de onda emitidas por los leds de luz roja e infrarroja y que trabajan a 660nm y 880nm respectivamente, la corriente de estos se configura entre 0 a 50 mA para este caso se define en la programación de Arduino IDE a 8mA y 40 mA, la precisión del valor obtenido depende de la resolución con 16 bits y ancho del pulso que determina el tiempo de

iluminación de los leds con 1600us, la velocidad de muestreo del fotodiodo es de 1500 muestras por segundo con una frecuencia a 100Hz. En la figura 23 se observa el código de configuración y acondicionamiento de sensor MAX30100.

```
#define REPORTING_PERIOD_MS      1500
#define PULSE_WIDTH              MAX30100_SPC_PW_1600US_16BITS
#define IR_LED_CURRENT           MAX30100_LED_CURR_40MA
#define LED_CURRENT              MAX30100_LED_CURR_20_8MA
#define SAMPLING_RATE            MAX30100_SAMPRATE_100HZ

void setup()
{
  Serial.begin(9600);
  u8g2.begin();
  initial_display();
  pox.begin();

  pox.setOnBeatDetectedCallback(onBeatDetected);
  pox.setIRLedCurrent(LED_CURRENT);
  sensor.setMode(MAX30100_MODE_SPO2_HR);
  sensor.setLedsPulseWidth(PULSE_WIDTH);
  sensor.setSamplingRate(SAMPLING_RATE);
}
```

Figura 23: Código de acondicionamiento sensor MAX30100.

Fuente: Elaborado por la investigadora

Comunicación Bluetooth

La comunicación Bluetooth se realiza mediante el módulo Bluetooth HC-05. Este utiliza el estándar de comunicación IEEE 802.15.1 compatible con el IoT. Este funcionará a una tasa de transferencia de 9600 baudios y se utilizará para la transmisión de datos de la siguiente manera:

```
Serial.print(HRclean);
Serial.print(',');
Serial.print(SpO2);
Serial.print(',');
tsLastReport = millis();
```

La tarjeta del Arduino Pro mini 3.3 tiene una conexión en sus pines analógicos hacia el sensor MAX30100 en sus pines SCL (reloj de sincronía) y SDA(datos) correspondiente para comunicación I2C y que es compartida hacia la pantalla, mientras que sus pines de transmisión y recepción están conectados hacia el módulo Bluetooth HC05. Para que el propietario y médico pueda observar los datos procesados que brinda la pechera, se añadió la pantalla OLED de 0.91 pulgadas, con una resolución de 128x34 pixeles y comunicación I2C. En el Anexo B, Anexo C se describen las características principales y los pines del Max30100 y la pantalla OLED respectivamente. En la figura 24 se detalla las conexiones de los componentes a la tarjeta Arduino Pro mini.

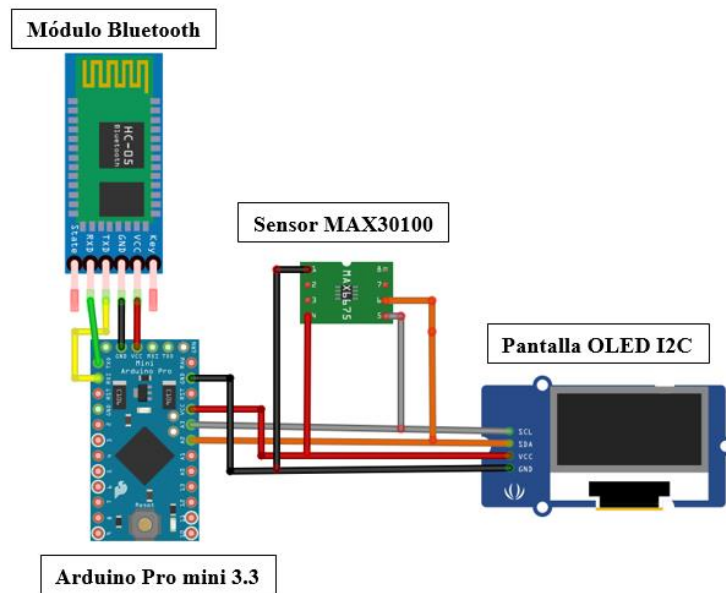


Figura 24: Diagrama de conexión de los elementos de la pechera portable.
Fuente: Elaborado por la investigadora

Suministro de energía de la pechera

El sistema de suministro de energía debe tener la capacidad de alimentar al sensor MAX30100, al Arduino Pro mini 3.3 y a la pantalla OLED con un rango de voltaje aproximado de 3.3V a 5V, se utiliza una batería de polímero de litio de 3,7 V con una capacidad de 550 mAh, se utiliza el módulo TP4056 para que el usuario pueda cargar el dispositivo mediante un cable USB micro. En la figura 22 se observa la conexión realizada para el sistema de suministro de energía de la pechera portable.

Para el suministro de energía se utilizó el módulo TP4056 y una batería de polímero de litio de 3.7 volts con 550 mAh. El circuito integrado posee una entrada para la batería, indicadores de carga, circuitos de protección de carga y salida para el sistema que se desea suministrar energía. Cuenta con una entrada de tipo USB mini para poder conectar el dispositivo y cargar la batería de manera adecuada, cuidando su vida útil. La distribución de los circuitos integrados fue colocada dentro de una case, realizado con filamento PLA o material de impresión 3D. El dispositivo portable permite la adquisición de signos vitales, el procesamiento adecuado y confiabilidad para el manejo de los datos. En la figura 25 se observa el diagrama de conexión para el suministro de energía del dispositivo.

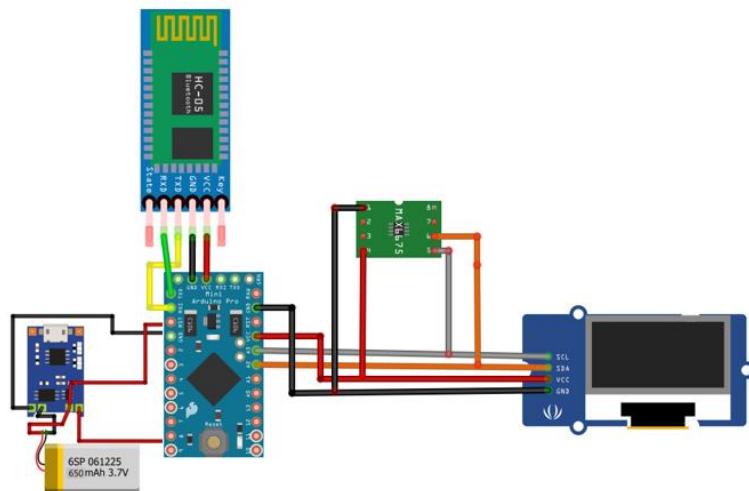


Figura 25: Diagrama de conexión para el suministro de energía del circuito.

Fuente: Elaborado por la investigadora

Para la visualización del nivel de energía del dispositivo se realiza un método condicional al medir los niveles de voltaje y relacionarlos un led colocado de manera física.

```
void loop()
{
  lectura = analogRead(A0);
  volt = lectura /1023 * 5.0;
  if (volt<=3.1){
    digitalWrite(ledR, LOW);
    digitalWrite(ledA, HIGH);
    digitalWrite(ledV, HIGH);
  }
  else if(volt>=3.5 && volt<=3.6){
    digitalWrite(ledR, HIGH);
    digitalWrite(ledA, LOW);
    digitalWrite(ledV, HIGH);
  }
  else if(volt>=3.9){
    digitalWrite(ledR, HIGH);
    digitalWrite(ledA, HIGH);
    digitalWrite(ledV, LOW);
  }
}
```

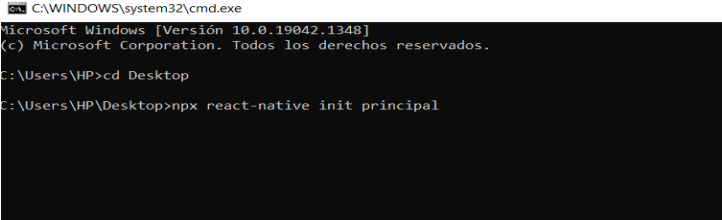
3.3.2 Gestión de información y monitoreo de datos

Entorno de programación

Después de adquirir los datos mediante el collar, los datos se transmiten vía comunicación inalámbrica con el módulo Bluetooth y son receptados por una aplicación móvil instalada en este caso en un dispositivo Android y desarrollada bajo el entorno de React Native.

Creación de la Aplicación en React Native

Se inicia creando el archivo con el siguiente comando “*npx react-native init principal*” por medio de cmd de Windows. En la figura 26 se observa la creación del archivo.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\HP>cd Desktop
C:\Users\HP\Desktop>npx react-native init principal
```

Figura 26: Creación de la aplicación en React Native.

Fuente: Elaborado por la investigadora

A continuación, se inicia programando en Visual Studio Code la página principal de la aplicación, para este caso la aplicación inicia con un splash interactivo con el médico y usuario. En el anexo H se muestra la programación de esta pantalla y en la figura 27 se muestra la pantalla que visualiza tanto el médico como el usuario.

Splash.js



Figura 27: Programación del Splash de bienvenida en React Native.

Fuente: Elaborado por la investigadora

De igual manera para la identificación de ingreso ya sea médico o propietario de mascotas se visualiza un splash y botones de ingreso respectivo, anexo H, código en React Native. En la figura 28 se visualiza la página de inicio de sesión al sistema.

Profile.js

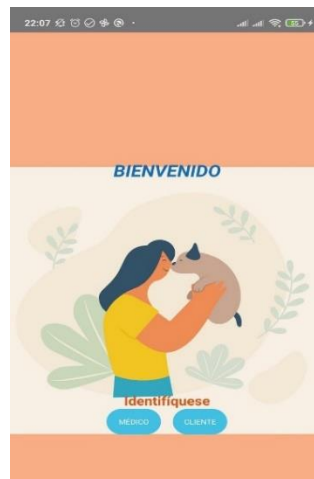


Figura 28: Página de inicio principal programado en React Native.

Fuente: Elaborado por la investigadora

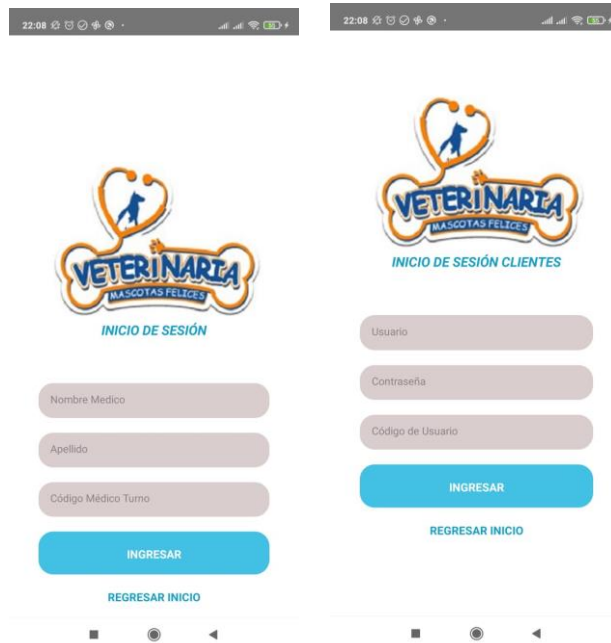


Figura 29: Pantallas de inicio de sesión para médico y clientes.

Fuente: Elaborado por la investigadora

Dentro de la programación en React Native es necesario registrar los datos del médico en turno y de los usuarios en este caso el propietario de las mascotas para ello se requiere de un sistema de logeo o de inicio de sesión para cada caso, teniendo en cuenta la creación de un sistema de bank-end y front-end, donde el back-end es la parte de desarrollo del elemento web que no es visible para el usuario mientras que el front-end es toda programación visible y que interactúa con el usuario. Para el caso de logeo de usuario se implementa un back-end desarrollado en Express que es un marco de desarrollo minimalista para Node.js permitiendo estructurar una aplicación con funcionalidades como enrutamiento, gestión de sesiones entre otras características, en esta sección se utiliza Express para hacer la conexión con la base de datos alojada en Centos 7 dentro de Amazon y se configura y programa dentro del archivo principal de la aplicación desarrollada. El código desarrollado en React Native para el servidor Express se detalla en el anexo H. En la figura 30 se observa el inicio de sesión exitoso por el médico veterinario.

Express

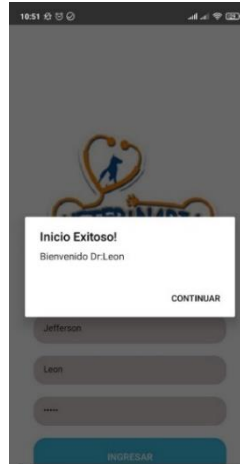


Figura 30: Consulta a la base de datos mediante Express y logeo exitoso.

Fuente: Elaborado por la investigadora

Después del logeo exitoso se inicia un menú con un nuevo Splash o imágenes en movimiento de bienvenida al sistema, esto se puede ver en la figura 31 y 32, donde el médico puede administrar y gestionar de mejor manera la información de sus pacientes, así como editar y eliminar el mismo, registrar a un nuevo paciente y visualizar el registro de vacunas de todos los pacientes.

Bienvenida al sistema para el médico veterinario.

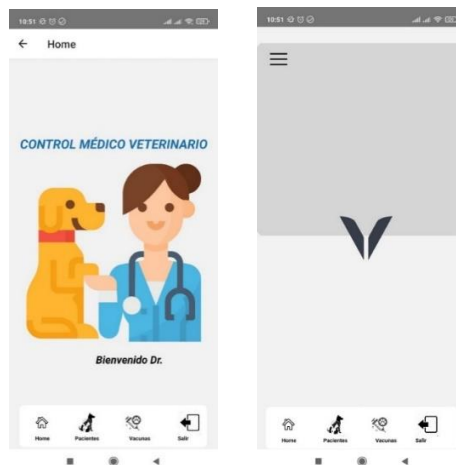


Figura 31: Creación de la aplicación en React Native.

Fuente: Elaborado por la investigadora

Express

```
app.get('/', (req, res) => {
  const query = "SELECT * FROM user";
  db.query(query, function(error, results, fields){
    res.json(results)
  });
});
```

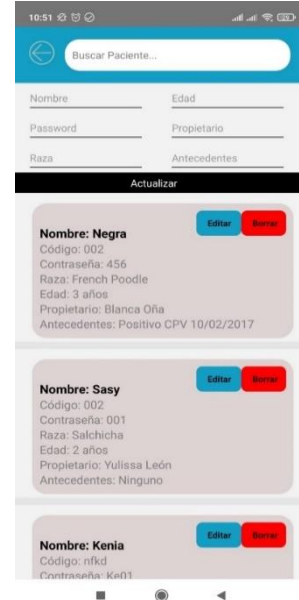


Figura 32: Pantalla de consulta de pacientes en React Native.

Fuente: Elaborado por la investigadora

Para el registro de los pacientes se utiliza el método POST dentro de la programación en Express. En la figura 33 se observa la pantalla de registro de pacientes.

Express

```
app.post('/', (req, res) => {
  const obj = req.body
  const query = "INSERT INTO user SET ?";
  db.query(query, obj, function(error, results, fields){
    res.json(results.insertId)
  });
});
```

Registro.js

```
async guardar(event) {
  event.preventDefault();
  try{
    let {username, password, codigo, raza, edad, antecedentes, propietario}= this.state;
    const obj={username, password, codigo, raza, edad, antecedentes, propietario}
    const res=await axios.post('http://192.168.101.15:4000/', obj);
    console.log(res.data)
    var hola=username;
    Alert.alert('Registro Exitoso!', 'Paciente: '+ hola, [
      {text: 'Continuar', onPress: () => console.log('ya esta presionado')}
    ])
  } catch(error){console.log(error)}
```

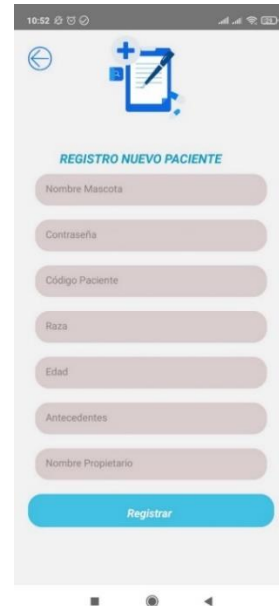


Figura 33: Pantalla de registro de pacientes en React Native.

Fuente: Elaborado por la investigadora

Finalmente en el menú para el médico tratante se puede observar el registro de vacunas de cada uno de sus pacientes y en donde se puede actualizar y colocar un nuevo registro de vacunas, de igual manera se realiza por medio del método UPDATE en Express hacia la base de datos. En la figura 34 se observa la pantalla de ingreso de datos sobre vacunación de pacientes.

Express



Figura 34: Pantalla de ingreso de datos de vacunación en React Native.

Fuente: Elaborado por la investigadora

La interfaz para el usuario es el mismo en cuanto al menú de bienvenida con un splash diferente para interactuar con el cuidado de las mascotas, en esta sección el usuario puede visualizar el carnet virtual de su mascota así como la fecha próxima de su siguiente vacuna y todos los datos informativos, mismos que han sido agregados con anterioridad por el médico veterinario en la etapa de actualización de carnet de vacuna y por último pero no menos importante se tiene la conexión con la pchera mediante

conexión vía inalámbrica Bluetooth para visualizar los datos que están siendo sensados. En la figura 35 se visualiza la bienvenida al sistema del usuario.

Bienvenida al sistema para el propietario de la mascota (usuario).



Figura 35: Pantalla de bienvenida al usuario en React Native.

Fuente: Elaborado por la investigadora

Para la programación del carnet de vacunación virtual del paciente igualmente se utiliza Express mediante el método POST para obtener los datos de la base de datos haciendo peticiones a la URL asignada en la figura 36 se puede ver el carnet virtual.

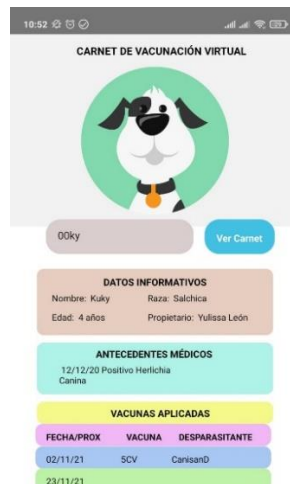


Figura 36: Carnet virtual en React Native.

Fuente: Elaborado por la investigadora

Y en la visualización de la frecuencia cardiaca de la mascota y la saturación de oxígeno en la sangre de igual manera nos conectamos a la Base de datos de Amazon mediante el puente Express y visualizamos los datos, principalmente se instaló la librería correspondiente a Bluetooth para react native “*react-native-bluetooth-serial-next*” y mediante programación en JavaScript se configuraron métodos y funciones de manera asíncrona para conectar, vincular y desconectar la interacción Bluetooth entre el módulo HC-05 y el Bluetooth del dispositivo Android y muestra la lista de dispositivos vinculados como se observa en la figura 37.

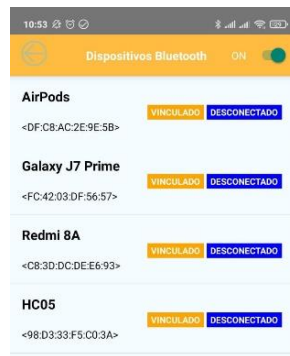


Figura 37: Lista de dispositivos bluetooth vinculados

Fuente: Elaborado por la investigadora

Las funciones tipo flecha para la conexión y desconexión de la comunicación Bluetooth ayudaron a que sea posible llamar desde cualquier parte del código, en la figura 38 se observa la pantalla de conexión al módulo Bluetooth de la pechera.



Figura 38: Pantalla de conexión a dispositivo Bluetooth.

Fuente: Elaborado por la investigadora

Lectura de los datos enviados por el módulo HC-05 mediante el siguiente código visualizado en la figura 39.

```
read = async () => {
  try {
    const data = await BluetoothSerial.readFromDevice();
    var arrayDeCadenas = data.split(',');
    this.setState({ data2: arrayDeCadenas[0]});
    this.setState({ HR: arrayDeCadenas[1]});
    var strReverse = arrayDeCadenas.reverse();
    this.setState({ data3: strReverse[0]});
    this.setState({ SPO2: strReverse[1]});
    const hola='34';
    this.setState({ id: hola});
    this.guardar();
  } catch (e) {
    console.log('Error', e);
  }
};
```

Figura 39: Código de lectura de datos del bus serial del módulo HC-05

Fuente: Elaborado por la investigadora

Visualización de los datos proporcionados por la pechera medidora de signos vitales, datos que se están enviando en tiempo real a la base de datos con la fecha exacta de transmisión. Se utilizó código en Express para insertar los datos de frecuencia cardiaca y de saturación de oxígeno en conjunto con la fecha de transmisión y el ID respectivo a cada mascota. En la figura 40 se observa los datos recibidos en tiempo real.



Figura 40: Código de insertar datos en base de datos y visualización en pantalla.

Fuente: Elaborado por la investigadora

3.1.3 Alojamiento en la nube

Se propone alojar los datos y el software de administración en la nube que proporciona Amazon con sus servicios llamados AWS (Servicios Web de Amazon), que proporciona 30 GB de almacenamiento y que son gratuitos durante 12 meses. Además, este servicio de nube posee aplicabilidad con el IoT, ya que posee varios servicios de administración de datos en este campo. El servicio gratuito de Amazon también permite seleccionar varios tipos de sistemas operativos virtuales como Centos, para la gestión de bases de datos y de ambientes virtuales, en donde son alojados los softwares de administración de datos. También brinda un servicio de DNS y de protocolos de comunicación TCP/IP y HTTP que serán necesario para el funcionamiento del sistema. Por demás, se propone este servicio ya que el investigador posee experiencia en su manejo, tanto del sistema operativo Centos 7 y del panel de administración de AWS. Los datos recopilados por los sensores son alojados en un servidor Express programado de manera local dentro de la configuración de React Native proporcionando un puente a la conexión con la Base de Datos MYSQL instalada en la instancia de Centos 7 de Amazon Web Services. En la figura 41 se observa los componentes que forman el servidor en la nube donde se desarrolló la interfaz web para el sistema electrónico.

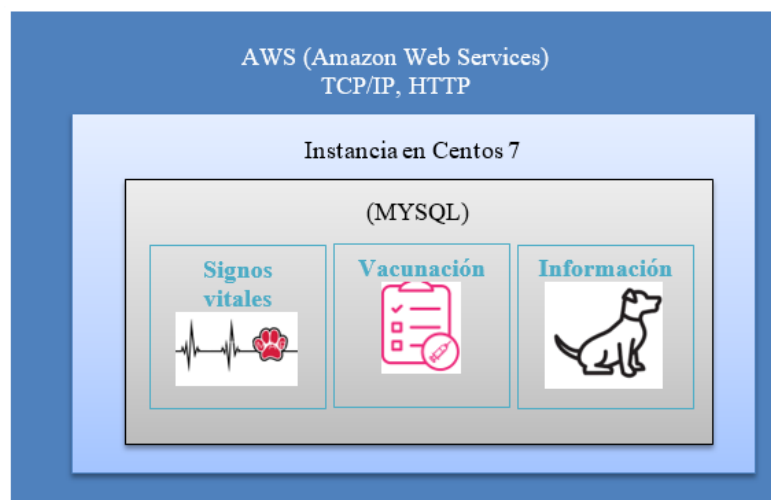


Figura 41: Componentes del servidor en la nube

Fuente: Elaborado por la investigadora

Base de datos MYSQL

MYSQL es instalada dentro de una instancia virtual alojada en la nube de Amazon Web Services y compatible para aplicaciones IoT, esta base de datos llamada Almacenamiento está configurada en MYSQL con una tabla inicial llamada users en donde contiene 3 tablas llamadas user, médico y medición, en donde user son los datos de todas las mascotas registradas, la tabla médicos son todos los médicos registrados y la tabla medición es los datos recopilados y almacenados por la pechera. En la figura 42 se puede observar el diagrama UML correspondiente a la base de datos, tablas e información que se almacena dentro del servidor alojado en la nube.

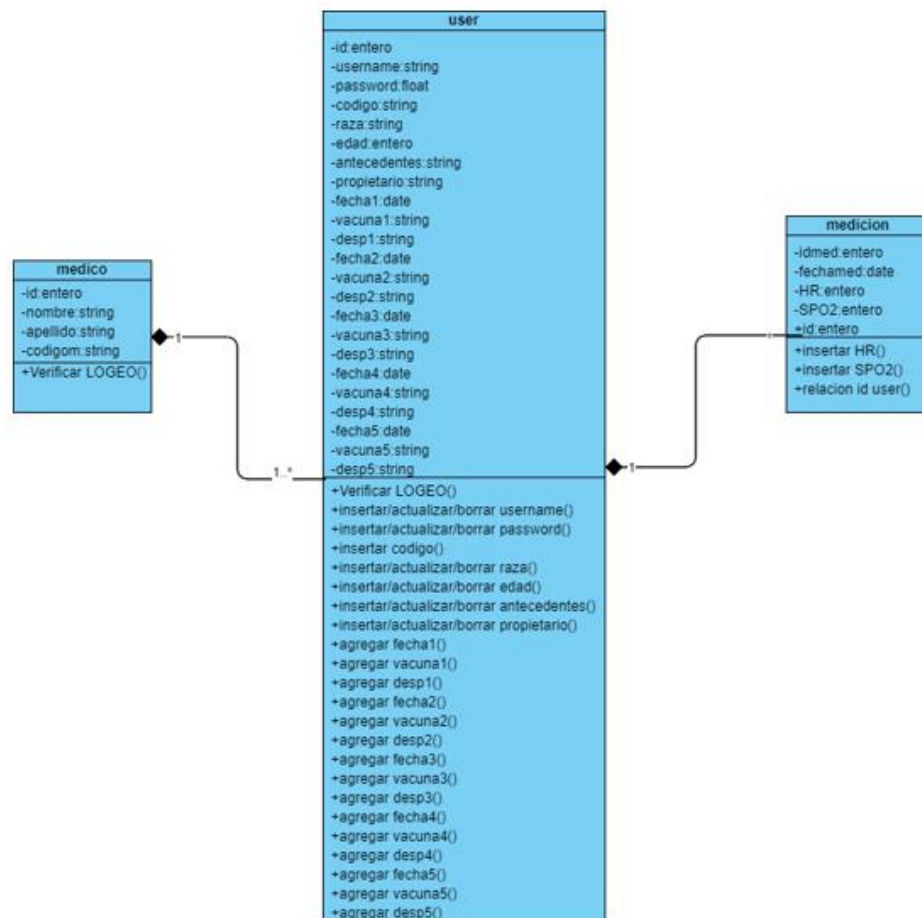


Figura 42: Diagrama UML de Base de datos.

Fuente: Elaborado por la investigadora

Tabla user

| id | username | password | codigo | raza | edad | antecedentes | propietario | fecha1 | vacuna1 | desp1 | fecha2 |
|----|----------|----------|--------|-----------------|---------|------------------------------------|----------------|----------|----------|----------|----------|
| 31 | Negra | 456 | 002 | French Poodle | 3 años | Positivo CPV 10/02/2017 | Blanca Oña | 02/10/21 | SCV Plus | Aprax | 22/10/21 |
| 32 | Sasy | 001 | 002 | Salchicha | 2 años | Ninguno | Yulissa León | 01/11/21 | SCV | Aprax | 22/11/21 |
| 33 | Kenia | Ke01 | nfkdl | French | 5 meses | Ninguno | Daniel Montero | 04/11/21 | SCV | Aprax | 25/11/21 |
| 34 | Kuky | 012 | 00ky | Salchicha | 4 años | 12/12/20 Positivo Herlichia Canina | Yulissa León | 02/11/21 | SCV | CanisanD | 23/11/21 |
| 35 | Dogy | dg01 | 00dg1 | Husky siberiano | 2 años | Ninguno | Ana Franco | (null) | (null) | (null) | (null) |
| 36 | Nash | na1 | 00nas1 | Pastor alemán | 1 año | Ninguno | Andrés Herrera | (null) | (null) | (null) | (null) |

Tabla médicos

| id | nombre | apellido | codigom |
|----|-----------|----------|---------|
| 2 | Jefferson | Leon | 00JL1 |

Tabla medición

| idmed | fehamed | HR | SPO2 | id |
|-------|---|----|------|----|
| 352 | Mon Nov 15 2021 17:12:19 GMT-0500 (-05) | HR | SPO2 | 35 |
| 353 | Mon Nov 15 2021 17:12:34 GMT-0500 (-05) | HR | SPO2 | 35 |
| 354 | Mon Nov 15 2021 17:12:41 GMT-0500 (-05) | HR | SPO2 | 35 |
| 355 | Mon Nov 15 2021 17:12:48 GMT-0500 (-05) | HR | SPO2 | 35 |

Enfermedades cardíacas en perros

En el capítulo 1 se menciona lo importante que es para este proyecto poder determinar con este dispositivo de manera temprana enfermedades en mascotas, es por ello que se debe conocer el rango de los valores a medir en cuanto a frecuencia cardíaca y saturación de oxígeno e identificar cuáles son las enfermedades que conllevan valores anormales en la medición de los mismos, las enfermedades cardíacas en canes son una de ellas y se definen como cardiopatía se refiere a aquellas enfermedades o problemas anatómicos que afectan al corazón del animal, y que como consecuencia empeoran la calidad de vida e incluso, en casos extremos pueden provocar la muerte.

De este modo, la frecuencia cardíaca de los perros se establece tomando en cuenta los valores referenciados en la tabla 6, los mismos valores que son necesarios para generar las respectivas alertas en el sistema de monitoreo.

Tabla 6. Frecuencia cardiaca con relación a la edad en perros [22]

| Edad | Frecuencia cardiaca (Bpm) |
|------------------------|----------------------------------|
| Cachorros (1-12 meses) | 110-120 |
| Adultos (1-9 años) | 90-100 |
| Ancianos (> 10 años) | 70-80 |

Asistente en Asterisk

Asterisk es una plataforma de telefonía de código abierto que convierte un ordenador en un servidor de comunicaciones VOIP, este servidor es capaz de crear una centralita IP. Diseñada originalmente para sistemas GNU/Linux es por eso que, en este proyecto, Asterisk está alojado en una instancia Centos 7 de Amazon Web Services y configurada de cierta manera que las extensiones de usuarios puedan comunicarse a través del internet, siendo compatible también con el internet de las cosas. La configuración de Asterisk también cuenta con la funcionalidad de conectarse mediante AGI con MYSQL o base de datos relaciones de Amazon la misma que cuenta con los datos recopilados por el collar y los administrados por la aplicación móvil. En la figura 43 se observa la estructura de la asistencia virtual mediante el software Asterisk hasta la interacción con el propietario de la mascota.



Figura 43: Estructura del sistema de teleasistencia

Fuente: Elaborado por la investigadora

Dentro de Asterisk se desarrolla un asistente virtual simulando una central telefónica, las llamadas que se realizan son a través de Internet y no solo de manera local es decir que cualquier usuario puede realizar consultas sobre su mascota, hasta el médico tratante sobre sus pacientes. Se desarrolló sobre un alojamiento dentro del servicio de instancias de Amazon bajo el sistema operativo Centos 7 con un logeo a través de un par de claves RSA navegando con la ayuda de Bitvise SSH ingresando por el puerto 22 y con la dirección pública generada en la creación de la instancia, a continuación, la instalación de Asterisk es sencilla ya que se requieren ciertos comandos que se han comprendido en clases prácticas dentro del estudio universitario. En la figura 44 se muestra la instancia creada y el ingreso mediante Bitvise SSH mediante el puerto 22.

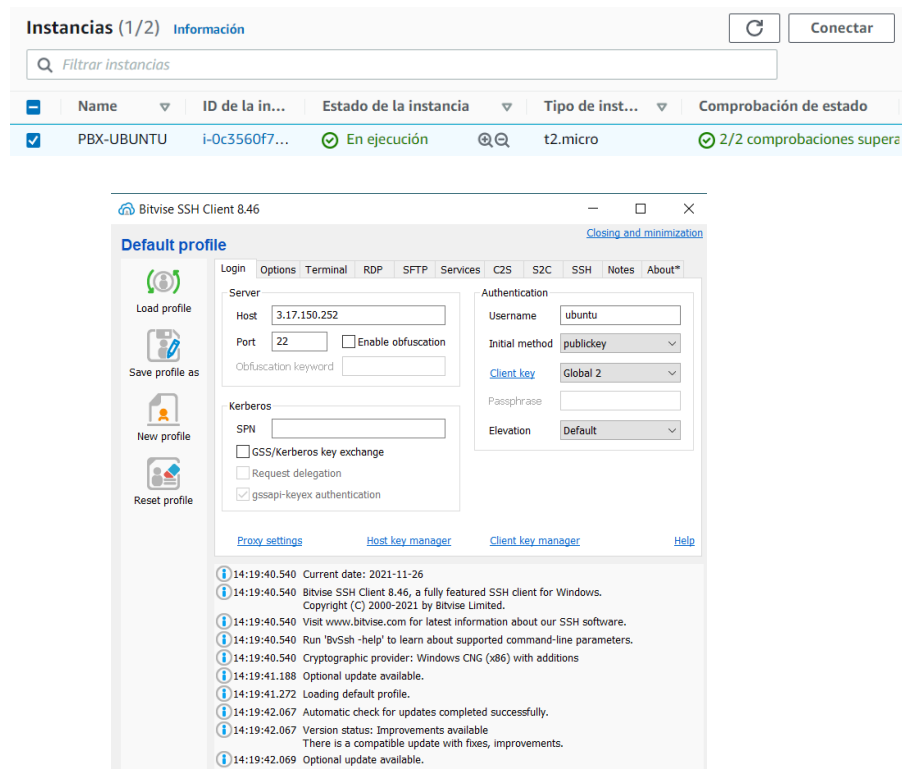


Figura 44: Ingreso a la instancia de Centos 7.

Fuente: Elaborado por la investigadora

Dentro de la configuración de las extensiones creadas para la central telefónica en Asterisk, se ha creado un menú IVR mediante la ayuda de google text to speech en

donde se transforma el texto en audio original de google, en el anexo I, se muestra los códigos implementados en esta sección.

3.4 Implementación final del proyecto

Al iniciar cada una de las actividades de este proyecto se realizaron pruebas de funcionamiento del circuito de la pechera portable para medición de signos vitales, en un circuito de pruebas dentro de una protoboard. Los datos que el sensor pulsioximétrico brindó eran reales y dentro del rango óptimo de manera constante existiendo pequeñas variaciones hasta estabilizarse. En la figura 45 se visualiza la placa de circuito impreso con todos los elementos soldados y en su case impreso en 3D para su posterior ensamblaje en la pechera.

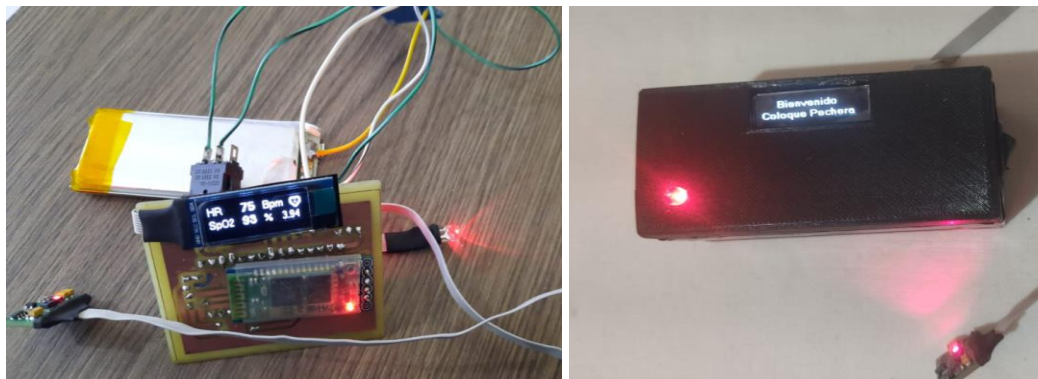


Figura 45: Circuito impreso y su case de protección.

Fuente: Elaborado por la investigadora

Después para la construcción del dispositivo se acopló adecuadamente el sensor, la placa Arduino, el sistema de suministro de energía, módulo Bluetooth y pantalla OLED dentro de una pechera con una correcta ubicación con el fin de ser portable para la mascota facilitando la adquisición de datos, este acople se puede visualizar en la figura 46.



Figura 46: Pechera portable ensamblada.

Fuente: Elaborado por la investigadora

El envío de datos a través del módulo bluetooth se lo verificó con un terminal bluetooth dentro del dispositivo Android teniendo conexión exitosa, al momento de iniciar la programación de la aplicación móvil en React Native se inició analizando las características del mismo indagando en varios repositorios y sitios oficiales ya que este software y entorno de programación es relativamente nuevo en el ámbito de creación de aplicaciones móviles funcionales.

La implementación continuó con la conexión entre la base de datos MYSQL y React requiriendo un marco de aplicación back-end o proceso invisible para el usuario, ejecutándose en segundo plano e interactuando en el logeo de sesión y consulta, actualización e inserción de datos hacia la base de datos llamada “Almacenamiento”.

La base de datos instalada en Centos 7 llamada ‘Almacenamiento’ tiene una capacidad de 9437184 bytes en disco con un almacenamiento para cada tabla de 16384 bytes teniendo la capacidad de crear 576 tablas dentro de la misma, para el caso de este proyecto se crearon 3 tablas (medición, médicos y user). En la figura 47 se visualiza las bases de datos creadas en la instancia como las tablas creadas dentro de la base de datos ‘Almacenamiento’.

| Base de datos | Tamaño (MB) | Tables_in_Almacenamiento |
|--------------------|-------------|--------------------------|
| Almacenamiento | 0.06 | medicion |
| information_schema | 0.07 | medicos |
| mysql | 0.63 | user |
| performance_schema | 0.00 | |

Figura 47: Base de datos creadas en el sistema y tablas de ‘Almacenamiento’.

Fuente: Elaborado por la investigadora

La capacidad de creación de filas para cada tabla se define por su capacidad o longitud de datos determinada en bytes, a continuación, se detalla las características de cada una de las tablas:

- La tabla médicos tiene como motor de base de datos a InnoDB que hace referencia a almacenamiento de datos de código abierto para MYSQL para todas sus distribuciones desde la versión 4.0, la versión instalada para este caso es la 10 con un formato compacto y con un número de filas de 1 ya que se tiene registrado a 1 médico veterinario, con 16384 bytes a utilizar y cada fila en esta tabla ocupa alrededor de 17 bytes es decir que esta tabla llamada médicos puede alojar hasta 960 filas correspondientes a los datos de médicos veterinarios. En la figura 48 se observa el detalle completo de la tabla médicos como el tamaño calculado de cada fila.

| Name | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length |
|---------|--------|---------|------------|------|----------------|-------------|-----------------|
| medicos | InnoDB | 10 | Compact | 1 | 16384 | 16384 | 0 |

| Index_length | Data_free | Auto_increment | Create_time | Update |
|--------------|-----------|----------------|---------------------|--------|
| 0 | 9437184 | 2 | 2021-11-29 17:43:20 | NULL |

| TamañoFila |
|------------|
| 17 |

Figura 48: Características generales tabla ‘médicos’.

Fuente: Elaborado por la investigadora

- Para el caso de la tabla user la capacidad es la misma a la tabla médicos pero con diferentes campos considerando las vacunas con sus respectivas fechas y desparasitaciones, es aquí en donde se estima la longitud en bytes de cada fila obteniendo como resultado 165 bytes utilizados aproximadamente en cada una de ellas con relación a la capacidad que es 16384 bytes, determinando así que la tabla user puede alojar hasta 100 filas es decir que el sistema puede abarcar hasta 100 usuarios en esta tabla teniendo la posibilidad de expandirse en las otras 573 tablas sin asignación en otras palabras un límite entre 57000 y 57300 usuarios. En la figura 49 se observa la descripción y detalle de la tabla ‘user’.

| Name | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length |
|------|--------|---------|------------|------|----------------|-------------|-----------------|
| user | InnoDB | 10 | Compact | 2 | 8192 | 16384 | 0 |

| Index_length | Data_free | Auto_increment | Create_time | Update_ti |
|--------------|-----------|----------------|---------------------|-----------|
| 0 | 9437184 | 5 | 2021-12-04 15:22:00 | NULL |

| TamañoFila |
|------------|
| 165 |

Figura 49: Características generales tabla ‘user’.

Fuente: Elaborado por la investigadora

- Concluyendo con la última tabla llamada ‘medición’ en donde se almacenan los datos correspondientes a signos vitales de las mascotas, de igual manera esta tabla cuenta con una capacidad de 16384 bytes, motor de base de datos InnoDB y se estima de acuerdo a la longitud de cada fila que es de 50 bytes es decir que en esta tabla se pueden crear 320 filas o datos de las mediciones realizadas por la pechera portable teniendo un alcance con las 573 tablas sin asignación de aproximadamente 187760 filas. En la figura 50 se observa las características de la tabla ‘medición’.

| Name | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length |
|----------|--------|---------|------------|------|----------------|-------------|-----------------|
| medicion | InnoDB | 10 | Compact | 2 | 8192 | 16384 | 0 |

| Index_length | Data_free | Auto_increment | Create_time | Update_time |
|--------------|-----------|----------------|---------------------|-------------|
| 0 | 9437184 | 12 | 2021-11-29 23:45:54 | NULL |

| TamañoFila |
|------------|
| 50 |

Figura 50: Características generales tabla ‘medición’.

Fuente: Elaborado por la investigadora

A continuación, se logró obtener el puente entre la aplicación y la base de datos alojada en servidor Centos en Amazon Web Services se procede a verificar la navegación de las pantallas, funcionalidades correspondientes a cada una de ellas y visualización con un diseño claro y preciso. Seguidamente se utiliza la librería “react-native-bluetooth-serial-next”, del módulo Bluetooth para poder conectar el dispositivo Android hacia la pechera donde está configurado el módulo HC-05 y poder también visualizar los datos obtenidos mediante indicadores métricos.

Culminando la etapa de adquisición y visualización de los datos dentro de la aplicación es hora de abordar el tema del servidor VOIP dentro de Asterisk, instalada en Centos 7 y utilizando el CLOUD de Amazon Web Services donde se instalaron todas las dependencias de Asterisk con sus respectivos códecs de audio,

se inició creando la instancia dentro de Amazon siendo Centos 7 el sistema operativo más apropiado, ya que no contaba con errores de lectura de paquetes y dependencias de instalación, obteniendo comunicación entre dos puntos estableciendo una llamada a través del Internet. A continuación, se realiza el menú IVR o menú de respuesta interactiva de voz en donde se utiliza como sonido la lectura de texto googles desarrollada por Google para aplicaciones Android con soporte para el idioma español, bajo la extensión 105 se inicia e ingresa la comunicación virtual donde empieza con un mensaje de bienvenida para posterior mencionar el menú de consultoría de vacunas, consultoría de signos vitales y finaliza con la opción de llamar a atención al cliente.

Para la generación de alertas del sistema se utilizó un archivo ejecutable .php con python para que al momento de que en la pechera se registre un valor fuera del rango normal de medición que se analizó en el capítulo 2 en cuanto a enfermedades cardiacas en perros, este realice una llamada automáticamente al usuario informándole sobre lo acontecido.

Abarcando el tema de asistencia virtual mediante llamadas telefónicas utilizando una centralita Asterisk es importante conocer los falsos positivos que genera nuestro servidor ante posibles ataques de intrusos o virus que tratan de ingresar al sistema provocando procesos de lentitud y latencia dentro del mismo, es por ello que se decidió brindar seguridad al sistema Asterisk instalando y configurando 'Fail2Ban' que es un analizador de logs y busca intentos fallidos de registro que mediante un script bloquea las direcciones IP provenientes de dichos ataques.

Dentro del servidor Asterisk es importante conocer el nivel de rendimiento de la centralita es por ello que se decide implementar un test de estrés dentro de Asterisk mediante la aplicación 'Sipp' que tiene como principal objetivo generar llamadas automáticas en la central del modo INVITE y BYE, en pocas palabras se establece una llamada cada cierto tiempo, Asterisk recibe esta llamada y muchas más y las libera poco a poco, conociendo la potencia o rendimiento de la central, para este caso se realizó un escenario por defecto Sipp con 5 llamadas simultáneas a la

extensión 7001 durante 10000 segundos y en la figura 51 se visualiza los resultados y la pantalla de estadísticas del test.

```
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
10.0(0 ms)/1.000s 5061 260.29 s 45 0.0.21.230:5060(UDP)

0 new calls during 0.787 s period 0 ms scheduler resolution
5 calls (limit 5) Peak was 5 calls, after 0 s
1 Running, 12 Paused, 7 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
INVITE -----> 45 220 40 0
100 <----- 0 0 0 0
180 <----- 0 0 0 0
183 <----- 0 0 0 0
200 <----- E-RTD1 0 0 0 0
ACK -----> 0 0
Pause [ 0ms] 0 0
BYE -----> 0 0 0 0
200 <----- 0 0 0 0

----- Test Terminated -----
```

```
----- Statistics Screen ----- [1-9]: Change Screen --
Start Time | 2022-01-26 04:03:52:328 | 1643169832.328668
Last Reset Time | 2022-01-26 04:08:11:839 | 1643170091.839931
Current Time | 2022-01-26 04:08:12:627 | 1643170092.627940
-----
```

| Counter Name | Periodic value | Cumulative value |
|-----------------------|----------------|------------------|
| Elapsed Time | 00:00:00:788 | 00:04:20:299 |
| Call Rate | 0.000 cps | 0.173 cps |
| Incoming call created | 0 | 0 |
| OutGoing call created | 0 | 45 |
| Total Call created | | 45 |
| Current Call | 5 | |
| Successful call | 0 | 40 |
| Failed call | 0 | 0 |
| Response Time 1 | 00:00:00:000 | 00:00:00:000 |
| Call Length | 00:00:00:000 | 00:00:31:511 |

```
----- Test Terminated -----
```

Figura 51: Escenario creado y tabla de estadísticas ‘Sipp’

Fuente: Elaborado por la investigadora

Es decir que la centralita Asterisk soporta 40 llamadas simultáneas sin algún tipo de interferencia, fallos o retardos en la comunicación.

Las pruebas finales fueron en conjunto con la aplicación y la pechera portable por la mascota. En el anexo J se visualizan la construcción del case de dispositivo con sus características de diseño. En la figura 52 se observa la pechera o dispositivo medidor de signos vitales siendo utilizado por la mascota.



Figura 52: Pechera midiendo signos vitales.

Fuente: Elaborado por la investigadora

En las figuras 53 y 54 se visualiza los diagramas de flujo del funcionamiento tanto del hardware como software del sistema.

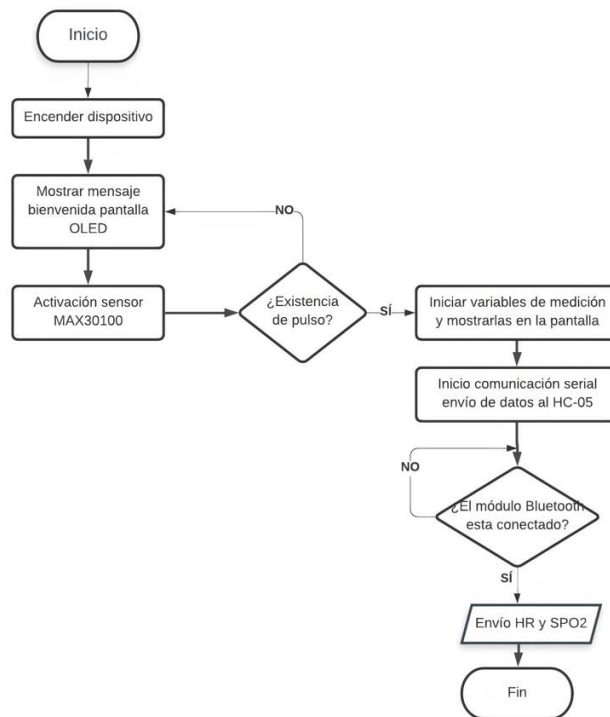


Figura 53: Diagrama de flujo para hardware del sistema.

Fuente: Elaborado por la investigadora

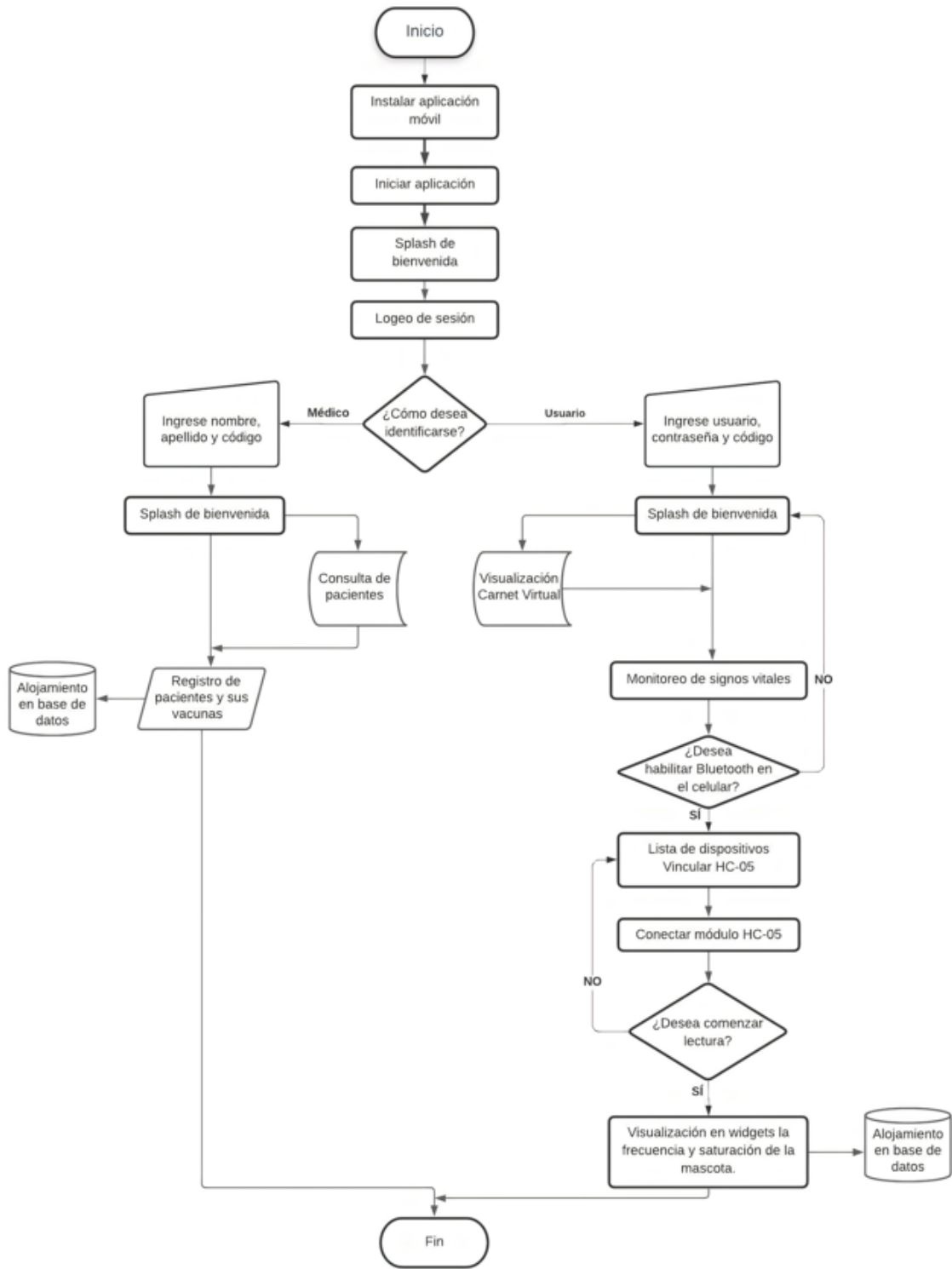


Figura 54: Diagrama de flujo para el software del sistema.

Fuente: Elaborado por la investigadora

3.5 Verificación de hipótesis

3.5.1 Pruebas de funcionamiento

Para las pruebas de funcionamiento, en la etapa de adquisición de datos se monitorearon los signos vitales de 3 pacientes de especie canina, raza pequeña de 4, 2 y 3 años respectivamente, se realizó una comparación entre un dispositivo medidor de signos vitales homologado para equipamiento médico en centros veterinarios denominado “HOLTER” y el sistema de monitoreo de signos vitales para mascotas desarrollado en este proyecto. Para ello se realizaron mediciones reales de signos vitales en los canes dentro de la clínica veterinaria “Mascotas Felices” utilizando dispositivos de índole profesional que registran valores de frecuencia cardíaca y saturación de oxígeno en la sangre para poder comparar los registros con el dispositivo de monitoreo de mascotas desarrollado en este proyecto.

DATOS E HISTORIAL MÉDICO DE LOS PACIENTES

El primer can utilizado para este proyecto de investigación lleva por nombre ‘Kuky’, tiene 4 años de edad, raza Salchicha, sexo hembra, peso 5 kg y es paciente rutinario de la Clínica Veterinaria Mascotas Felices, cuenta con un antecedente médico en el cual sufrió de parálisis en sus patas traseras a causa de Erliquia canina, de acuerdo a esta condición sus revisiones son constantes ya que esta enfermedad provoca daño en diversos órganos como es hígado, riñones y corazón, por lo que se convierte en un paciente de análisis indicado para este proyecto. En la figura 55 se observa una de las revisiones del can Kuky en donde se utiliza el método de auscultación con el dispositivo médico HOLTER obteniendo valores un tanto elevados ya que la mascota reconoce que está en un lugar desconocido.



Figura 55: Medición de signos vitales por auscultación dentro del centro veterinario can Kuky.

Fuente: Elaborado por la investigadora

Para realizar el análisis de comparación del dispositivo homologado para procedimientos médicos denominado 'HOLTER' y el sistema de monitoreo realizado en este proyecto se recurrió a realizar esta medición en el hogar del paciente ya que se encuentra en un lugar conocido por ende se tranquilizan y es más fácil observar si se presenta alguna variación y se puede observar en la figura 56.



Figura 56: Medición de signos vitales HOLTER y dispositivo en la pechera.

Fuente: Elaborado por la investigadora

El segundo can en analizar lleva por nombre 'Nena', tiene 2 años de edad, raza Mestizo, sexo hembra, peso 8 kg y es paciente rutinario de la Clínica Veterinaria Mascotas Felices, cuenta con un antecedente médico en el cual sufrió de Parvovirus Canino, virus que lo padeció cuando era cachorra y favorablemente supo recuperarse, esta enfermedad desgarró los intestinos provocando sangrado al momento de hacer sus deposiciones y vómitos constantes afectando nuevamente hígado, riñones y corazón, de acuerdo a esta condición sus revisiones son constantes ya que debe también cumplir con su carnet de vacunación y con sus chequeos, por lo que la convierte también en un paciente de análisis indicado para este proyecto. En la figura 57 se observa una de las revisiones del can Nena en donde se utiliza el método de auscultación con el dispositivo médico HOLTER obteniendo valores un tanto elevados ya que la mascota reconoce que está en un lugar desconocido.



Figura 57: Medición de signos vitales por auscultación con Nena dentro del centro veterinario.

Fuente: Elaborado por la investigadora

De igual manera que con el primer can analizado para realizar la comparación del dispositivo homologado para procedimientos médicos denominado 'HOLTER' y el sistema de monitoreo realizado en este proyecto se recurrió a realizar esta medición

en el hogar del paciente ya que se encuentra en un lugar conocido por ende se tranquilizan y es más fácil observar si se presenta alguna variación y se puede observar en la figura 58.

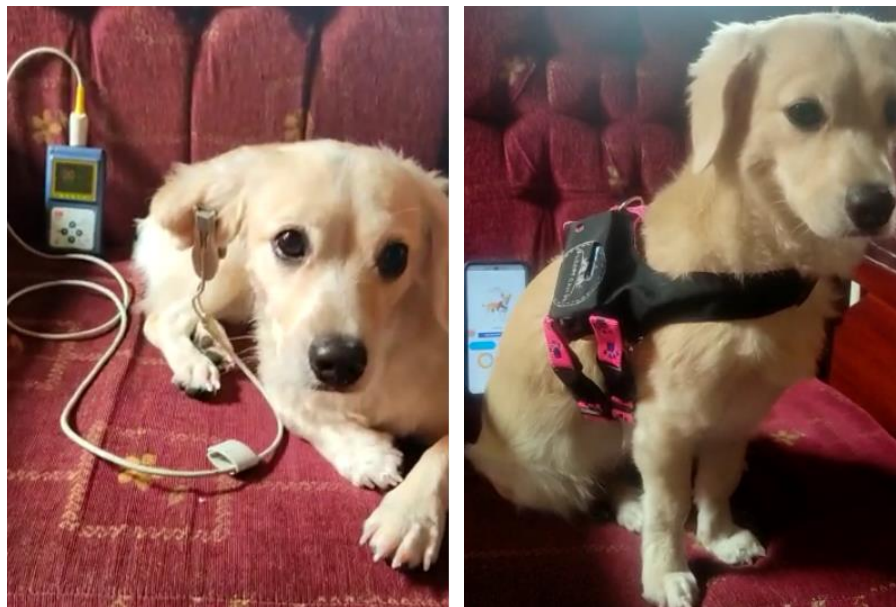


Figura 58: Medición de signos vitales HOLTER y dispositivo en la pechera.

Fuente: Elaborado por la investigadora

El tercer can en analizar lleva por nombre ‘Ojito’, tiene 3 años de edad, raza French Poodle, sexo hembra, peso 7 kg y es paciente rutinario de la Clínica Veterinaria Mascotas Felices, cuenta con un antecedente médico en el cual sufrió de Infección por gusanera en uno de sus ojos, enfermedad que lamentablemente hizo que perdiera uno de ellos, esta situación hizo que su sistema inmune se debilitara tanto que estos parásitos llegaron hasta el cerebro, es uno de los casos más preocupantes dentro de la veterinaria, es por ello que elige a este can para este proyecto ya que en sus chequeos constantes podemos visualizar si existe alguna anomalía desencadenante de la enfermedad que sufrió. En la figura 59 se observa una de las revisiones del can Ojito en donde se utiliza el método de auscultación con el dispositivo médico HOLTER obteniendo valores un tanto elevados ya que la mascota reconoce que está en un lugar desconocido.



Figura 59: Medición de signos vitales por auscultación con Ojito dentro del centro veterinario.

Fuente: Elaborado por la investigadora

De igual manera que con el primero y segundo can analizado para realizar la comparación del dispositivo homologado para procedimientos médicos denominado 'HOLTER' y el sistema de monitoreo realizado en este proyecto se recurrió a realizar esta medición en un lugar más tranquilo por ende es más fácil observar si se presenta alguna variación y se puede observar en la figura 60.

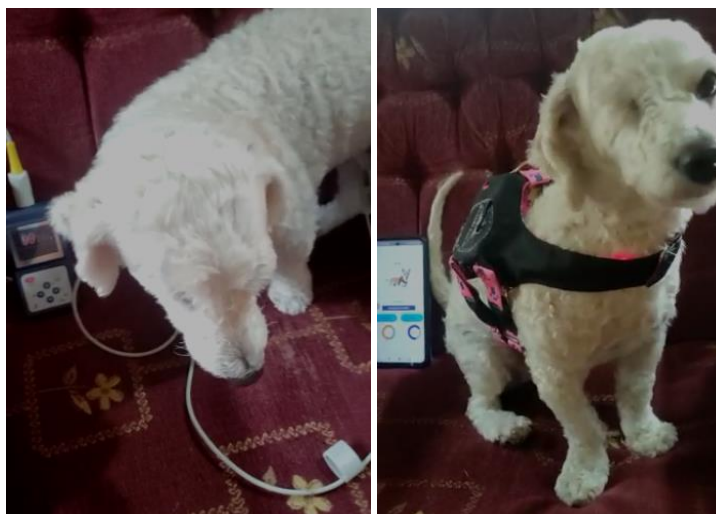


Figura 60: Medición de signos vitales HOLTER y dispositivo en la pechera.

Fuente: Elaborado por la investigadora

A continuación, se registran los valores medidos por un instrumento médico HOLTER dentro del centro médico veterinario obteniendo datos elevados ya que los canes reconocen que se encuentran en un lugar desconocido en consecuencia se vuelven intranquilos, por ende se decidió realizar las tomas de este instrumento en su hogar ya que no se estresan o sienten intimidados junto con los valores registrados por la pechera, en la tabla 7, 9 y 11 se puede observar la comparación de los dispositivos en 3 días después de que el sensor se estabilizara se tomaron 10 lecturas seguidas de la frecuencia cardíaca o pulsaciones por minuto, mientras que en la tabla número 8, 10 y 12 de igual manera se recabaron los valores obtenidos en saturación de oxígeno en la sangre en 3 días de lecturas cuando el sensor estabilizó su lectura con 10 intentos. La medición de signos vitales de la pechera envía los datos a la nube mediante la aplicación móvil y su conexión al módulo Bluetooth.

Tabla 7. Mediciones BPMs paciente Kuky.
Elaborado por: La investigadora

| Intentos | Pulsaciones por minuto (BPM) Paciente Kuky | | | | | |
|-------------------|---|---------|--------------------|---------|--------------------|---------|
| | Día 1 | | Día 2 | | Día 3 | |
| | Instrumento Médico | Pechera | Instrumento Médico | Pechera | Instrumento Médico | Pechera |
| 1 | 136 | 137 | 135 | 136 | 137 | 134 |
| 2 | 136 | 136 | 135 | 136 | 137 | 135 |
| 3 | 135 | 135 | 135 | 135 | 137 | 135 |
| 4 | 135 | 135 | 135 | 136 | 136 | 131 |
| 5 | 134 | 134 | 134 | 135 | 136 | 133 |
| 6 | 134 | 133 | 135 | 135 | 135 | 132 |
| 7 | 134 | 133 | 134 | 135 | 135 | 134 |
| 8 | 135 | 134 | 134 | 134 | 135 | 132 |
| 9 | 134 | 134 | 134 | 133 | 136 | 133 |
| 10 | 134 | 134 | 134 | 133 | 135 | 134 |
| Promedio | 134,70 | 134,50 | 134,50 | 134,80 | 135,90 | 133,30 |
| Varianza | 0,68 | 1,61 | 0,28 | 1,29 | 0,77 | 1,79 |
| Desviación | 0,82 | 1,27 | 0,53 | 1,14 | 0,88 | 1,34 |

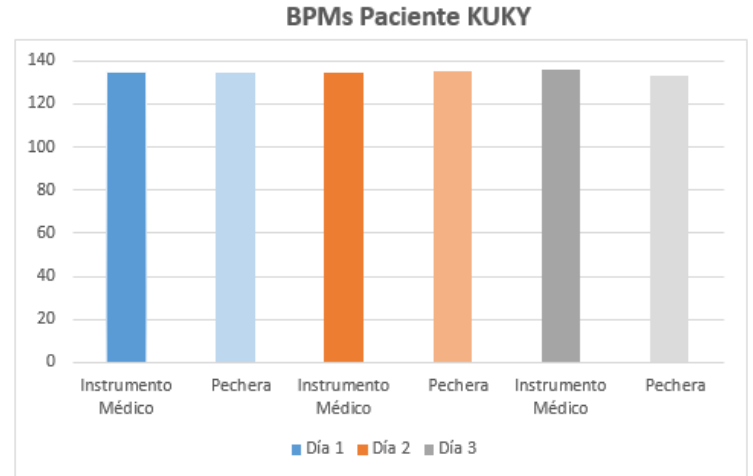


Figura 61: Diagrama de barras medición BPMs paciente Kuky.

Fuente: Elaborado por la investigadora

Tabla 8. Mediciones SPO2 paciente Kuky.

Elaborado por: La investigadora

| Intentos | Saturación de oxígeno en la sangre (SPO2 %) Paciente Nena | | | | | |
|-------------------|--|---------|--------------------|---------|--------------------|---------|
| | Día 1 | | Día 2 | | Día 3 | |
| | Instrumento Médico | Pechera | Instrumento Médico | Pechera | Instrumento Médico | Pechera |
| 1 | 98 | 93 | 98 | 96 | 99 | 96 |
| 2 | 98 | 94 | 98 | 95 | 99 | 95 |
| 3 | 97 | 96 | 98 | 94 | 99 | 95 |
| 4 | 97 | 93 | 98 | 94 | 99 | 94 |
| 5 | 97 | 93 | 98 | 96 | 99 | 95 |
| 6 | 97 | 93 | 97 | 95 | 99 | 95 |
| 7 | 97 | 94 | 97 | 94 | 99 | 96 |
| 8 | 97 | 94 | 97 | 96 | 99 | 99 |
| 9 | 97 | 95 | 97 | 96 | 99 | 99 |
| 10 | 97 | 94 | 97 | 96 | 99 | 99 |
| Promedio | 97,20 | 93,90 | 97,50 | 95,20 | 99,00 | 96,30 |
| Varianza | 0,16 | 0,89 | 0,25 | 0,76 | 0,00 | 3,41 |
| Desviación | 0,42 | 0,99 | 0,53 | 0,92 | 0,00 | 1,95 |

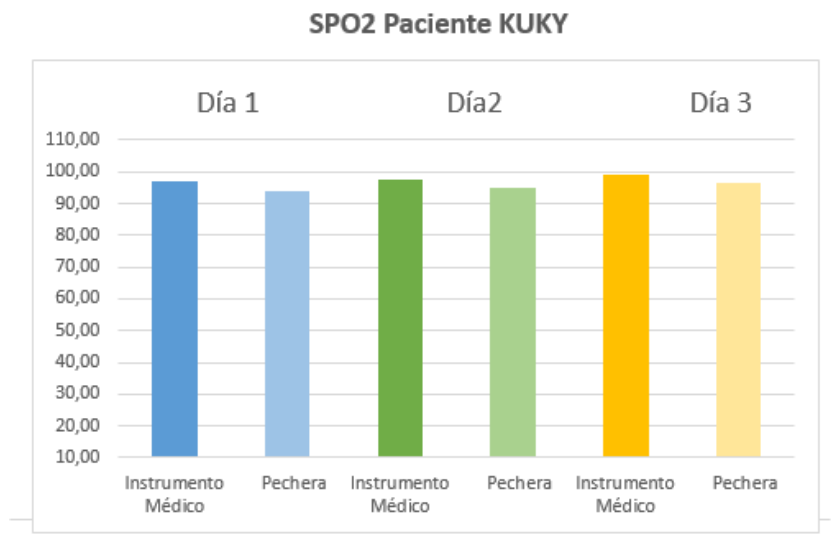


Figura 62: Diagrama de barras medición SPO2 paciente Kuky.

Fuente: Elaborado por la investigadora

Tabla 9. Mediciones BPMs paciente Nena.

Elaborado por: La investigadora

| Intentos | Pulsaciones por minuto (BPM) Paciente Nena | | | | | |
|-------------------|---|---------|--------------------|---------|--------------------|---------|
| | Día 1 | | Día 2 | | Día 3 | |
| | Instrumento Médico | Pechera | Instrumento Médico | Pechera | Instrumento Médico | Pechera |
| 1 | 83 | 84 | 85 | 83 | 86 | 85 |
| 2 | 84 | 83 | 84 | 84 | 85 | 86 |
| 3 | 82 | 84 | 85 | 85 | 84 | 84 |
| 4 | 81 | 80 | 84 | 84 | 85 | 83 |
| 5 | 80 | 80 | 84 | 83 | 84 | 83 |
| 6 | 80 | 82 | 84 | 84 | 84 | 82 |
| 7 | 80 | 83 | 83 | 84 | 83 | 83 |
| 8 | 81 | 86 | 83 | 86 | 83 | 85 |
| 9 | 80 | 80 | 83 | 83 | 83 | 84 |
| 10 | 81 | 83 | 83 | 83 | 83 | 86 |
| Promedio | 81,20 | 82,50 | 83,80 | 83,90 | 84,00 | 84,10 |
| Varianza | 1,76 | 3,65 | 0,56 | 0,89 | 1,00 | 1,69 |
| Desviación | 1,40 | 2,01 | 0,79 | 0,99 | 1,05 | 1,37 |

BPMs Paciente NENA

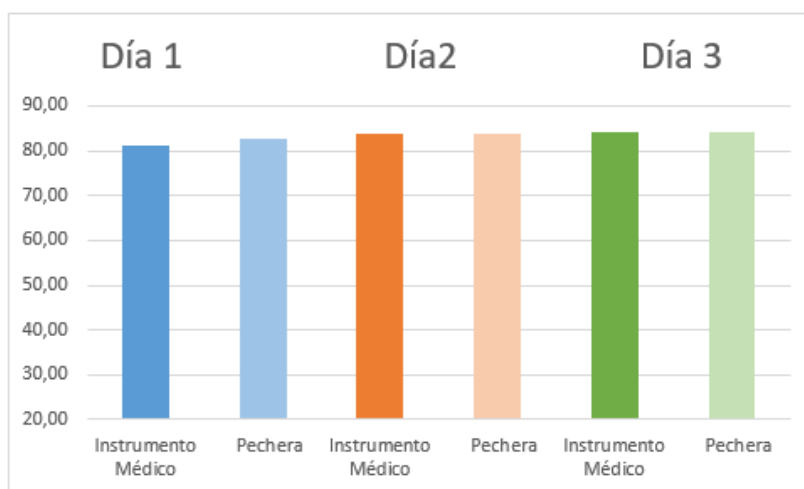


Figura 63: Diagrama de barras medición BPMs paciente Nena.

Fuente: Elaborado por la investigadora

Tabla 10. Mediciones SPO2 paciente Nena.

Elaborado por: La investigadora

| Intentos | Saturación de oxígeno en la sangre (SPO2) Paciente Nena | | | | | |
|-------------------|---|---------|--------------------|---------|--------------------|---------|
| | Día 1 | | Día 2 | | Día 3 | |
| | Instrumento Médico | Pechera | Instrumento Médico | Pechera | Instrumento Médico | Pechera |
| 1 | 98 | 99 | 98 | 96 | 99 | 96 |
| 2 | 98 | 98 | 98 | 95 | 99 | 97 |
| 3 | 98 | 98 | 98 | 96 | 99 | 97 |
| 4 | 99 | 99 | 98 | 96 | 99 | 98 |
| 5 | 99 | 99 | 98 | 96 | 99 | 98 |
| 6 | 99 | 99 | 97 | 97 | 99 | 97 |
| 7 | 99 | 98 | 97 | 97 | 99 | 98 |
| 8 | 99 | 98 | 97 | 98 | 99 | 99 |
| 9 | 99 | 99 | 97 | 98 | 99 | 99 |
| 10 | 99 | 99 | 97 | 98 | 99 | 99 |
| Promedio | 98,70 | 98,60 | 97,50 | 96,70 | 99,00 | 97,80 |
| Varianza | 0,21 | 0,24 | 0,25 | 1,01 | 0,00 | 0,96 |
| Desviación | 0,48 | 0,52 | 0,53 | 1,06 | 0,00 | 1,03 |

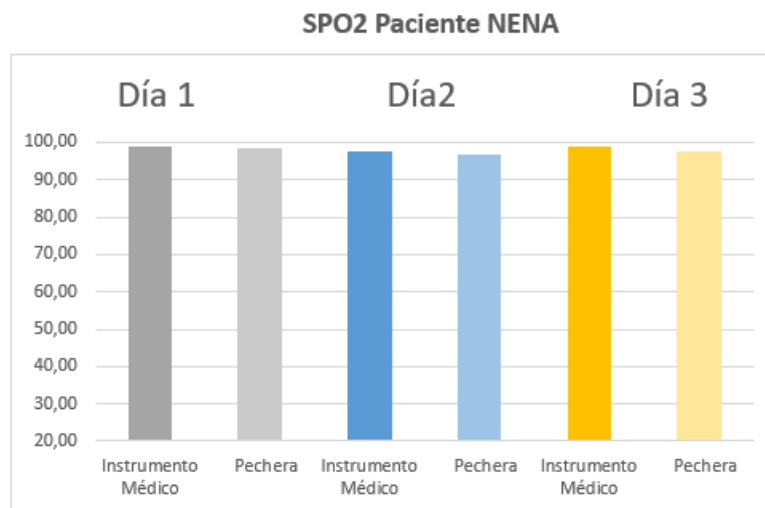


Figura 64: Diagrama de barras medición SPO2 paciente Nena.

Fuente: Elaborado por la investigadora

Tabla 11. Mediciones BPMs paciente Ojito.

Elaborado por: La investigadora

| Intentos | Pulsaciones por minuto (BPM) Paciente Ojito | | | | | |
|-------------------|--|---------|--------------------|---------|--------------------|---------|
| | Día 1 | | Día 2 | | Día 3 | |
| | Instrumento Médico | Pechera | Instrumento Médico | Pechera | Instrumento Médico | Pechera |
| 1 | 158 | 132 | 145 | 136 | 147 | 144 |
| 2 | 159 | 136 | 147 | 136 | 147 | 145 |
| 3 | 160 | 145 | 148 | 138 | 147 | 145 |
| 4 | 160 | 147 | 151 | 140 | 146 | 143 |
| 5 | 160 | 150 | 152 | 145 | 146 | 143 |
| 6 | 160 | 155 | 152 | 148 | 145 | 143 |
| 7 | 159 | 155 | 153 | 148 | 145 | 142 |
| 8 | 160 | 155 | 153 | 150 | 145 | 142 |
| 9 | 160 | 155 | 153 | 151 | 136 | 143 |
| 10 | 160 | 155 | 153 | 151 | 135 | 143 |
| Promedio | 159,60 | 148,50 | 150,70 | 144,30 | 143,90 | 143,30 |
| Varianza | 0,49 | 72,94 | 8,68 | 38,46 | 20,32 | 1,12 |
| Desviación | 0,70 | 8,54 | 2,95 | 6,20 | 4,51 | 1,06 |

BPMs Paciente OJITO

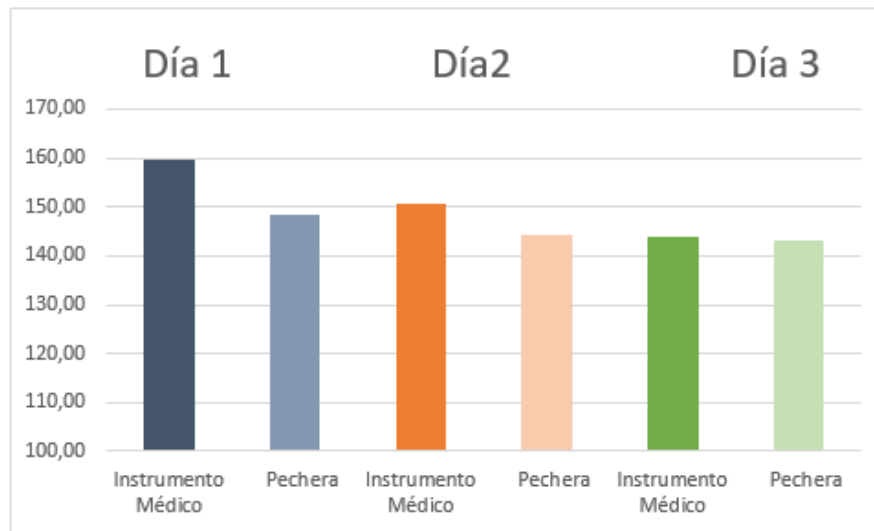


Figura 65: Diagrama de barras medición BPMs paciente Ojito.

Fuente: Elaborado por la investigadora

Tabla 12. Mediciones SPO2 paciente Ojito.

Elaborado por: La investigadora

| Intentos | Saturación de oxígeno en la sangre (SPO2) Paciente Ojito | | | | | |
|-------------------|--|---------|--------------------|---------|--------------------|---------|
| | Día 1 | | Día 2 | | Día 3 | |
| | Instrumento Médico | Pechera | Instrumento Médico | Pechera | Instrumento Médico | Pechera |
| 1 | 99 | 94 | 98 | 96 | 99 | 95 |
| 2 | 99 | 94 | 98 | 95 | 99 | 95 |
| 3 | 99 | 94 | 98 | 94 | 99 | 94 |
| 4 | 99 | 94 | 98 | 94 | 99 | 94 |
| 5 | 99 | 94 | 98 | 96 | 99 | 94 |
| 6 | 99 | 95 | 97 | 95 | 99 | 95 |
| 7 | 99 | 94 | 97 | 94 | 99 | 94 |
| 8 | 99 | 94 | 97 | 96 | 99 | 96 |
| 9 | 99 | 94 | 97 | 96 | 99 | 96 |
| 10 | 99 | 93 | 97 | 96 | 99 | 95 |
| Promedio | 99,00 | 94,00 | 97,50 | 95,20 | 99,00 | 94,80 |
| Varianza | 0,00 | 0,20 | 0,25 | 0,76 | 0,00 | 0,56 |
| Desviación | 0,00 | 0,47 | 0,53 | 0,92 | 0,00 | 0,79 |

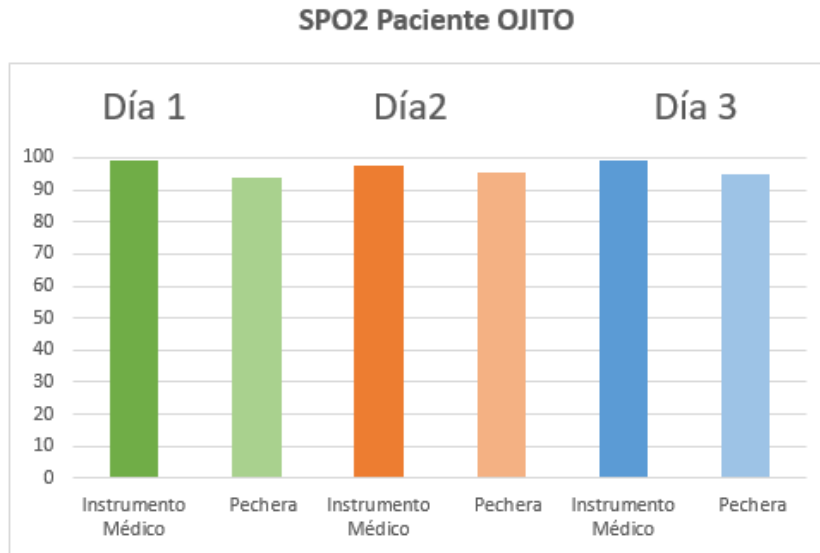


Figura 66: Diagrama de barras medición SPO2 paciente Ojito.

Fuente: Elaborado por la investigadora

Confiabilidad del dispositivo

Para el cálculo de la confiabilidad del dispositivo se utiliza la temática relacionada con estadística que se refiere al coeficiente de fiabilidad que es un instrumento de medida en donde se intenta establecer la precisión de medida de los dispositivos y relaciona el término que a mayor fiabilidad menor es el error de medida.

El método de Alfa Cronbach identifica la confiabilidad como consistencia interna donde el valor máximo de confiabilidad es 1 y los valores medidos determinados como coeficiente oscilatorio esta entre 0 y 1. [28]

Su fórmula es:

$$\alpha = \frac{K}{K-1} \left(1 - \frac{\sum V_i}{V_t} \right)$$

Donde:

- α : Coeficiente de Cronbach
- K : Número de muestras realizadas

- V_i : Varianza individual
- V_t : Varianza total

El valor a comparar en este apartado es la frecuencia cardiaca en el cual se utilizó la pechera conectado al sensor MAX30100 y un dispositivo HOLTER utilizado en centros médicos veterinarios para conocer estas magnitudes. El HOLTER generó un valor de confiabilidad de 0,83 entrando así en un intervalo Alto. Para la pechera se obtuvo un valor de alfa de 0,57 y esto ingresa en un intervalo de confiabilidad moderada, los datos recabados para realizar esta comparación y determinar la confiabilidad de los datos medidos por el prototipo diseñado en este proyecto se tomaron en un tiempo estimado de 3 días seguidos con la misma mascota, en el anexo D se muestra de manera detallada los cálculos realizados en cada instrumento de medida y en la figura 67 se observa la confiabilidad obtenida por el dispositivo HOLTER con las mediciones realizadas al can Kuky.

| CAN KUKY HOLTER | | | | |
|-------------------|-------------------|-------------------|-------------------|-----------------------|
| Número Mediciones | Instrumento Dia 1 | Instrumento Dia 2 | Instrumento Dia 3 | Instrumento BPMs Suma |
| 1 | 136 | 135 | 137 | 408 |
| 2 | 136 | 135 | 137 | 408 |
| 3 | 135 | 135 | 137 | 407 |
| 4 | 135 | 135 | 136 | 406 |
| 5 | 134 | 134 | 136 | 404 |
| 6 | 134 | 135 | 135 | 404 |
| 7 | 134 | 134 | 135 | 403 |
| 8 | 135 | 134 | 135 | 404 |
| 9 | 134 | 134 | 136 | 404 |
| 10 | 134 | 134 | 135 | 403 |
| Promedio | 134,7 | 134,5 | 135,9 | 405,1 |
| Varianza | 0,677777778 | 0,277777778 | 0,766666667 | 3,877777778 |
| Desviación | 0,823272602 | 0,527046277 | 0,875595036 | 1,969207398 |

| Ecuación a desarrollar | |
|--|-------------|
| $\alpha = \frac{K}{K-1} \left(1 - \frac{\sum V_i}{V_t} \right)$ | |
| Datos | |
| K | 3 |
| $\sum V_i$ | 1,722222222 |
| V_t | 3,877777778 |
| Alfa Cronbac | 0,833810888 |

Figura 67: Cálculo de confiabilidad dispositivo HOLTER.

Fuente: Elaborado por la investigadora

En la figura 68 se detalla el cálculo realizado para obtener el coeficiente de Cronbach de la pechera tanto para la frecuencia cardíaca y saturación de oxígeno en la sangre obteniendo valores dentro del rango de estimación moderado.

| | | | | | CAN KUKY | | | | | | | | | |
|-------------------|--------------------|--------------------|--------------------|-------------------|--|-------------|-------|--|---|---|------------|-------------|-------|-----|
| Número Mediciones | Pechera BPMs Dia 1 | Pechera BPMs Dia 2 | Pechera BPMs Dia 3 | Pechera BPMs Suma | | | | | | | | | | |
| 1 | 137 | 136 | 134 | 407 | Ecuación a desarrollar | | | | | | | | | |
| 2 | 136 | 136 | 135 | 407 | $\alpha = \frac{K}{K-1} \left(1 - \frac{\sum V_i}{V_t} \right)$ <table border="1"> <thead> <tr> <th colspan="2">Datos</th> </tr> </thead> <tbody> <tr> <td>K</td> <td>3</td> </tr> <tr> <td>$\sum V_i$</td> <td>4,688888889</td> </tr> <tr> <td>V_t</td> <td>7,6</td> </tr> </tbody> </table> | | Datos | | K | 3 | $\sum V_i$ | 4,688888889 | V_t | 7,6 |
| Datos | | | | | | | | | | | | | | |
| K | 3 | | | | | | | | | | | | | |
| $\sum V_i$ | 4,688888889 | | | | | | | | | | | | | |
| V_t | 7,6 | | | | | | | | | | | | | |
| 3 | 135 | 135 | 135 | 405 | | | | | | | | | | |
| 4 | 135 | 136 | 131 | 402 | | | | | | | | | | |
| 5 | 134 | 135 | 133 | 402 | | | | | | | | | | |
| 6 | 133 | 135 | 132 | 400 | | | | | | | | | | |
| 7 | 133 | 135 | 134 | 402 | | | | | | | | | | |
| 8 | 134 | 134 | 132 | 400 | | | | | | | | | | |
| 9 | 134 | 133 | 133 | 400 | | | | | | | | | | |
| 10 | 134 | 133 | 134 | 401 | | | | | | | | | | |
| Promedio | 134,5 | 134,8 | 133,3 | 402,6 | | | | | | | | | | |
| Varianza | 1,611111111 | 1,288888889 | 1,788888889 | 7,6 | Alfa Cronbach | 0,574561404 | | | | | | | | |
| Desviación | 1,269295518 | 1,135292424 | 1,33749351 | 2,75680975 | | | | | | | | | | |

| Número Mediciones | Pechera SPO2% Dia 1 | Pechera SPO2% Dia 2 | Pechera SPO2% Dia 3 | Pechera SPO2% Suma | | | | | | | | | | |
|-------------------|---------------------|---------------------|---------------------|--------------------|---|-------------|-------|--|---|---|------------|---|-------|------|
| 1 | 93 | 96 | 96 | 285 | Ecuación a desarrollar | | | | | | | | | |
| 2 | 94 | 95 | 95 | 284 | $\alpha = \frac{K}{K-1} \left(1 - \frac{\sum V_i}{V_t} \right)$ <table border="1"> <thead> <tr> <th colspan="2">Datos</th> </tr> </thead> <tbody> <tr> <td>K</td> <td>3</td> </tr> <tr> <td>$\sum V_i$</td> <td>5</td> </tr> <tr> <td>V_t</td> <td>7,84</td> </tr> </tbody> </table> | | Datos | | K | 3 | $\sum V_i$ | 5 | V_t | 7,84 |
| Datos | | | | | | | | | | | | | | |
| K | 3 | | | | | | | | | | | | | |
| $\sum V_i$ | 5 | | | | | | | | | | | | | |
| V_t | 7,84 | | | | | | | | | | | | | |
| 3 | 96 | 94 | 95 | 285 | | | | | | | | | | |
| 4 | 93 | 94 | 94 | 281 | | | | | | | | | | |
| 5 | 93 | 96 | 95 | 284 | | | | | | | | | | |
| 6 | 93 | 95 | 95 | 283 | | | | | | | | | | |
| 7 | 94 | 94 | 96 | 284 | | | | | | | | | | |
| 8 | 94 | 96 | 99 | 289 | | | | | | | | | | |
| 9 | 95 | 96 | 99 | 290 | | | | | | | | | | |
| 10 | 94 | 96 | 99 | 289 | | | | | | | | | | |
| Promedio | 93,9 | 95,2 | 96,3 | 285,4 | | | | | | | | | | |
| Varianza | 0,89 | 0,7 | 3,41 | 7,84 | Alfa Cronbach | 0,543367347 | | | | | | | | |
| Desviación | 0,94428926 | 0,918936583 | 1,946506843 | 2,951459149 | | | | | | | | | | |

Figura 68: Cálculo de confiabilidad pechera.

Fuente: Elaborado por la investigadora

Estos mismos cálculos de obtención de la confiabilidad de la pechera en cuanto a la medición de signos vitales como frecuencia cardíaca y saturación de oxígeno se realizaron para los canes de estudio Nena y Ojito los mismos se pueden visualizar en el anexo D.

Análisis de errores obtenidos

Los errores obtenidos se analizan de acuerdo a la práctica realizada centrándose en la estabilidad de lectura del sensor ya que el sujeto de análisis en este caso no es una persona que puede tener control sobre ella misma, sino todo lo contrario, las mascotas por su naturaleza reaccionan a agentes externos como al ruido, a personas extrañas, al clima, todo eso hace que el animal genere inestabilidad, al colocar la pechera en un animal en descanso y en la comodidad de su hogar la medición es al instante, esto no sucede cuando la mascota se siente incómoda o con temperatura o ladrando por algún agente externo a él, provocando cierta inestabilidad en la lectura del sensor MAX30100, lo mismo ocurre con el dispositivo médico denominado HOLTER, ya que sus lecturas varían de acuerdo a como está situado el animal en consecuencia estos dispositivos lo utilizan en gran parte para procedimientos quirúrgicos cuando el animal se encuentra completamente sedado. La inestabilidad en las lecturas se denominan errores iniciales y se determina con el porcentaje de error obtenido en los diferentes días en el cual se realizó las pruebas de funcionamiento, calculado en base al retardo en segundos(s) que tienen los intervalos de tiempo de cambios abruptos de valores en la pechera observados en la pantalla OLED y en la aplicación móvil instalada en el teléfono celular o interfaz de monitoreo.

Para el cálculo de porcentaje de error en los diferentes días se determinó realizar la medición en los 3 sujetos de análisis con un total de 5 pruebas, en donde se compara la medición obtenida por el dispositivo médico HOLTER y el dispositivo diseñado en este proyecto, el sensor MAX30100 requiere de una calibración o acondicionamiento para eliminar aquellos datos que no son reales y que provocan mediciones falsas. En las tablas 13, 14, 15, 16 y 17 se observan los valores de la frecuencia cardíaca y saturación de oxígeno obtenidos y su error porcentual referente a la medición hecha con el dispositivo comercial.

Tabla 13. Tabla de error porcentual referente a las mediciones obtenidas en el día 1.

Elaborado por: La investigadora

| Prueba 1 | FR (bpm) HOLTER | FR (bpm) Prototipo | Error (%) | SPO2 (%) HOLTER | SPO2 (%) Prototipo | Error (%) |
|----------|--------------------|-----------------------|--------------|--------------------|-----------------------|--------------|
| Kuky | 136 | 137 | 0.75 | 98 | 93 | 5.37 |
| Nena | 83 | 84 | 1.19 | 98 | 99 | 1.01 |
| Ojito | 158 | 155 | 1.93 | 99 | 97 | 2.06 |

Tabla 14. Tabla de error porcentual referente a las mediciones obtenidas en el día 2.

Elaborado por: La investigadora

| Prueba 2 | FR (bpm) HOLTER | FR (bpm) Prototipo | Error (%) | SPO2 (%) HOLTER | SPO2 (%) Prototipo | Error (%) |
|----------|--------------------|-----------------------|--------------|--------------------|-----------------------|--------------|
| Kuky | 135 | 136 | 0.74 | 98 | 96 | 2.08 |
| Nena | 85 | 83 | 2.4 | 98 | 96 | 2.08 |
| Ojito | 145 | 148 | 2.02 | 98 | 96 | 2.08 |

Tabla 15. Tabla de error porcentual referente a las mediciones obtenidas en el día 3.

Elaborado por: La investigadora

| Prueba 3 | FR (bpm) HOLTER | FR (bpm) Prototipo | Error (%) | SPO2 (%) HOLTER | SPO2 (%) Prototipo | Error (%) |
|----------|--------------------|-----------------------|--------------|--------------------|-----------------------|--------------|
| Kuky | 137 | 134 | 2.23 | 99 | 97 | 2.06 |
| Nena | 86 | 85 | 1.18 | 99 | 97 | 2.06 |
| Ojito | 146 | 143 | 2.09 | 99 | 95 | 4.21 |

Tabla 16. Tabla de error porcentual referente a las mediciones obtenidas en el día 4.

Elaborado por: La investigadora

| Prueba 4 | FR (bpm) HOLTER | FR (bpm) Prototipo | Error (%) | SPO2 (%) HOLTER | SPO2 (%) Prototipo | Error (%) |
|----------|--------------------|-----------------------|--------------|--------------------|-----------------------|--------------|
| Kuky | 136 | 135 | 0.74 | 99 | 98 | 1.02 |
| Nena | 87 | 86 | 1.16 | 98 | 99 | 1.01 |
| Ojito | 144 | 142 | 1.40 | 98 | 97 | 1.03 |

Tabla 17. Tabla de error porcentual referente a las mediciones obtenidas en el día 5.

Elaborado por: La investigadora

| Prueba 4 | FR (bpm) HOLTER | FR (bpm) Prototipo | Error (%) | SPO2 (%) HOLTER | SPO2 (%) Prototipo | Error (%) |
|----------|--------------------|-----------------------|--------------|--------------------|-----------------------|--------------|
| Kuky | 135 | 134 | 0.74 | 98 | 97 | 1.03 |
| Nena | 86 | 85 | 1.17 | 99 | 98 | 1.02 |
| Ojito | 143 | 140 | 2.14 | 98 | 98 | 1.02 |

De acuerdo a las pruebas realizadas en 5 días y en los 3 canes de estudio se logró comprobar que las mediciones realizadas por el prototipo diseñado se encuentran en el rango de funcionamiento adecuado comparado con el dispositivo HOLTER de uso comercial, ya que el porcentaje de error obtenido en cada lectura no sobrepasa del 3% que es el porcentaje máximo permitido para la homologación de dispositivos médicos según la Agencia Nacional de Regulación, Control y Vigilancia Sanitaria en Ecuador.

3.5.2 Costo del prototipo Precios de Hardware

Para este proyecto dentro del hardware que se adquirió para la pechera portable están los elementos electrónicos de medición, control y suministro de energía detallados en la tabla 18.

Tabla 18. Detalle de precios de hardware utilizado.
Elaborado por: La investigadora

| ítem | Detalle | Unidad | Cantidad | Valor Unitario | Total |
|------|----------------------------|--------|----------|-------------------------|----------|
| 1 | Batería LiPo 3.3 V 550 mAh | c/u | 1 | \$ 15,00 | \$ 15,00 |
| 2 | Cable de cobre | metros | 1 | \$ 0,25 | \$ 0,25 |
| 3 | Case (acrílico) | caja | 1 | \$ 15,00 | \$ 15,00 |
| 4 | Arduino Promini 3.3 | c/u | 1 | \$ 12,00 | \$12,00 |
| 5 | Estaño | metros | 2 | \$ 1,00 | \$ 2,00 |
| 6 | Interruptor | c/u | 1 | \$ 0,50 | \$ 0,50 |
| 7 | Programación App | c/u | 1 | \$ 20,00 | \$ 20,00 |
| 8 | Pantalla OLED I2C 0.91" | c/u | 1 | \$ 6,00 | \$ 6,00 |
| 9 | Sensor MAX30100 | c/u | 1 | \$ 9,00 | \$ 9,00 |
| 10 | Módulo Bluetooth HC-05 | c/u | 1 | \$ 6,50 | \$ 6,50 |
| 11 | TP40056 | c/u | 1 | \$ 2,50 | \$ 2,50 |
| | | | | Subtotal | \$81,55 |
| | | | | I.V.A (12%) | \$11,79 |
| | | | | Imprevistos (5%) | \$4,91 |
| | | | | Total | \$98,25 |

Precios de Software

Algunos programas utilizados para la realización de este proyecto son libres es decir que no incluyen precio al adquirirlos como son:

- Arduino IDE
- React Native
- Express

- Node JS
- Visual Studio Code

Mientras que los alojamientos en servidores virtuales tienen un costo que sus proveedores los determinan de acuerdo al tamaño que ocupen y las conexiones que pueden establecer, entre otros recursos más.

Amazon Web Services cuenta con las siguientes características y precios:

- La capa gratuita de AWS incluye 750 horas de instancias t2.micro de Linux y Windows, cada mes durante un año.
- Si el tamaño de la instancia supera el límite de 30GB para más horas el consumo por determinación de size o tamaño es de 0,0154 USD por hora, es decir que la operación mensual en de \$10,34 por la utilización de recursos en AWS.

Precio mano de obra

El precio de la mano de obra se determina de acuerdo al sueldo de un Ingeniero en Electrónica y Comunicaciones y los meses transcurridos para la culminación del proyecto. El sueldo de un ingeniero en este ámbito según el Ministerio de Trabajo es de \$430 en adelante y puede incrementar de acuerdo al rango en que se desempeñe y el tiempo en el que se realizó este proyecto es de 6 meses que multiplicando ambos valores se obtiene un precio de mano de obra de \$2580.

Precio del sistema de monitoreo de mascotas

De acuerdo al análisis y diseño del prototipo y el tiempo estimado para la implementación, con una ganancia del 30% se define junto con la “Teoría de Precios” de Milton Fridman la siguiente fórmula el precio de venta del producto:

$$\text{Precio de venta} = [(\text{costo del artículo}) \div (100 - \text{porcentaje de ganancia})] \times 100$$

$$\text{Precio de venta} = [(98,25) \div (100 - 30)] \times 100$$

$$\text{Precio de venta} = [(98,25) \div (100 - 30)] \times 100 = 140$$

El precio de venta del sistema de monitoreo de mascotas se evalúa entre los \$140 dólares.

3.5.3 Resultados

Como resultado de la realización de este proyecto, fueron datos similares a los reales en cuanto a medición de signos vitales en perros que definen confiabilidad del prototipo, datos visualizados en tiempo real sin ningún tipo de pérdida o latencia, comunicación ágil entre aplicación móvil, base de datos y servidor en la nube.

Portabilidad

La portabilidad del dispositivo medidor de signos vitales se define a la movilidad que puede tener y que se útil dentro de un determinado tiempo. De acuerdo a las pruebas de funcionamiento se ha determinado que la capacidad de carga que tiene el dispositivo y el rendimiento en horas es de 3 horas constantemente activo.

- **Autonomía de la batería utilizada**

Para determinar la duración en horas de la batería de nuestro sistema se utiliza la siguiente fórmula en donde se hace énfasis al voltaje e intensidad de la batería con la intensidad consumida de los elementos conectados a la misma:

La batería de alimentación lipo se alimenta con un voltaje de 3,7V y una corriente de 550mAH, el módulo de comunicación Bluetooth tiene un consumo de corriente máximo de 30mA, el sensor MAX30100 mediante la obtención de las variables de medición consumen 20mA de corriente, la visualización en pantalla OLED consumen 7mA en corriente como valor

máximo, el procesamiento de los datos en Arduino Promini es de 150mA, todos estos datos se pueden visualizar en sus respectivas hojas técnicas.

Cálculo de las horas de duración de la batería:

$$H = \frac{W_b}{W_c} = \left(\frac{V_b \times I_b}{V_b \times I_c} \right)$$
$$H = \frac{W_b}{W_c} = \left(\frac{3,7 \times 550}{3,7 \times 207} \right) = 2,8 \cong 3 \text{ horas}$$

Donde:

H : Duración de la batería. W_c : Potencia de consumo.

W_b : Potencia de la batería. I_b : Corriente de la batería.

I_c : Corriente de consumo.

Sistemas de monitoreo y alertas

Dentro del sistema de monitoreo y alertas se desarrollaron 2 formas para que la interacción con el usuario sea la más óptima, ágil y funcional. La aplicación móvil diseñada y programada en React Native puro es una de ellas, en esta aplicación como ya se ha mencionado, el médico puede administrar y gestionar de mejor manera la información de sus pacientes mientras que los propietarios pueden visualizar la información, vacunas y signos vitales de sus mascotas y tanto el médico como el propietario pueden hacer uso del asistente virtual, segundo método de monitoreo, por medio de llamadas telefónicas al servidor. Para el sistema de alertas se genera una llamada cuando un valor medido de la pechera esté fuera del rango aceptado como normal. Entre las funciones del sistema VOIP se encuentran:

- Bienvenida al sistema por medio del asistente virtual.
- Comandos de teclado para la comunicación con el asistente virtual.

- Información interactiva con el usuario sobre vacunación del paciente.
- Información interactiva sobre signos vitales registrados.
- Consulta a la base de datos MYSQL dentro del servidor Asterisk.
- Alertas generadas por el sistema virtual mediante llamadas telefónicas.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Al culminar con el trabajo de titulación se presenta las siguientes conclusiones:

- Se analizaron diferentes sistemas de monitoreo de mascotas que han sido desarrollados en el país y en el extranjero, así como una encuesta realizada a los médicos veterinarios indagando la forma en que manejan la información de sus pacientes y si estarían dispuestos a virtualizar su gestión ayudando a la asistencia médica de manera temprana. Todos los dispositivos miden en tiempo real pero no manejan la información receptada en servidores como lo hace este sistema.
- Se diseñó e implementó un sistema electrónico de monitoreo de mascotas dentro de un collar/pechera portable utilizando comunicación inalámbrica vía Bluetooth para la transmisión en conjunto con el desarrollo de un aplicación móvil disponible para Android y iOS programada en React Native en donde la gestión y administración de toda la información está disponible tanto como para el médico como para el usuario, para el monitoreo de signos vitales se utilizó la misma aplicación recabando los datos adecuadamente.
- Los datos registrados por el sensor y transmitidos por el módulo Bluetooth pasan por la aplicación desarrollada en React Native y directamente se alojan en el servicio de Base de datos MYSQL mediante un back-end implementado en Express, agilizando la comunicación ya que se ejecuta en segundo plano, es por ello que el usuario final ya sea médico o cliente tienen la capacidad de consultar datos de las mascotas sean estos de registro de vacunas o de signos vitales, mediante un sistema de consultoría virtual e interactivo instalado en Asterisk dentro de una instancia virtual de Amazon Web Services, el mismo que genera alertas cuando registra datos anormales en la medición.

- Las mediciones registradas son analizadas en comparación a un dispositivo médico homologado denominado HOLTER, determinando que el máximo error porcentual registrado no sobrepasa del 3% que es el porcentaje máximo permitido para la homologación de dispositivos médicos según la Agencia Nacional de Regulación, Control y Vigilancia Sanitaria en Ecuador y la confiabilidad del dispositivo en todos los casos es moderada de acuerdo al rango establecido por el coeficiente de Cronbach que es desde 0.4 hasta 0.6.

4.2 Recomendaciones

En la realización de este trabajo de titulación se han determinado algunas recomendaciones a seguir:

- Para lograr depurar correctamente la aplicación desarrollada en React Native es necesario instalar los SDK compatibles con Android y lograr que se puedan realizar cambios sin necesidad de volver a cargar el código fuente.
- Para una adecuada lectura de datos por medio del sensor MAX30100 se recomienda colocar directamente a la piel en donde el pulso sea más fuerte, para el caso de las mascotas en el cuello cerca de las arterias principales.
- Para el desarrollo de aplicaciones dentro de un lenguaje de programación relativamente nuevo es indispensable tener conocimientos de programación básica para poder entender las estructuras, métodos, funciones y relacionar el renderizado de cada componente de React Native, el uso de librerías es crucial para el adecuado funcionamiento de la aplicación ya que la mayoría de funcionalidades no son propias, ya que son adaptaciones del desarrollo de aplicaciones web.

BIBLIOGRAFÍA

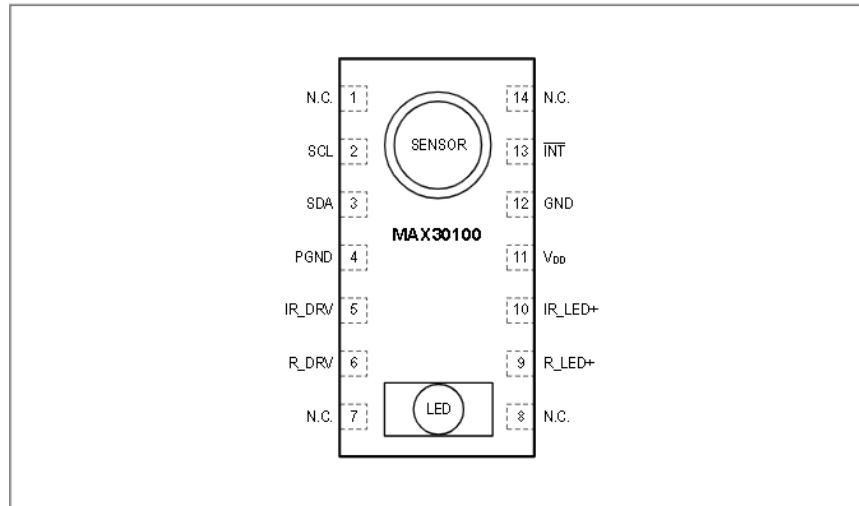
- [1] E. B. S. a. R. J. Istiadi, «Consultation Services Using Ivrr Telephony Based On Expert System Perspective,» IEEE, USA, 2015.
- [2] A. V. a. P. K. Archana Gupta, «Tele Health Therapy: An Ambient Technology,» IEEE, USA, 2015.
- [3] S. K. a. N. Sadiq, «Desing And Configuration of Voip Based Pbx Using Asterisk Server and Opnet Platform,» IEEE, USA, 2016.
- [4] J. C.-M. I. B. A. F. Andrei Tchernykh, «Configurable Cost-quality Optimization of Cloud-Based Voip,» Jornal of Parallel and Distributed Computing, USA, 2018.
- [5] E. Williams, Z. Cleghern y D. Roberts, «A Smart Collar for Assessment of Activity Levels and Environmental Conditions for Guide Dogs*,» IEEE, Canadá, 2020.
- [6] J. Castañeda, «Estudio de Planificación, Construcción y Equipamiento del Hospital,» Quito, 2014.
- [7] B. Emma, «Statista,» Statista, 10 10 2018. [En línea]. Available: <https://www.statista.com/statistics/808374/share-of-pet-owners-with-electronic-tracking-devices-for-their-pets-by-species-us/>. [Último acceso: 10 05 2021].
- [8] J. Estrada, M. Calva, A. Rodríguez y C. Tipantuña, «Seguridad de la Telefonía IP en Ecuador: Análisis en Internet,» Universidad Técnica Equinoccial, Quito-Ecuador, 2016.
- [9] T. Adriana, «Universidad UTE,» 01 04 2016. [En línea]. Available: <http://repositorio.ute.edu.ec/handle/123456789/17291>. [Último acceso: 15 10 2020].

- [10] L. Valdez y C. Tejada, «Alltime IoT: Tecnología para la Educación,» Bogotá, Colombia, 2018.
- [11] G. Santiago y T. Juan, «Salud Animal,» Curso de Producción Animal, España, 2001.
- [12] A. Alonso, «Aplicaciones móviles en salud: potencial, normativa de seguridad y regulación,» Revista Cubana de Información en Ciencias de la Salud, vol. 28, n° 3, 2017.
- [13] NeoAttack, «NeoAttack,» 06 03 2021. [En línea]. Available: <https://neoattack.com/neowiki/entorno-de-desarrollo/>. [Último acceso: 01 01 2022].
- [14] L. Ricardo, «React Native: acortando las distancias entre desarrollo y diseño móvil multiplataforma,» Revista Digital Universitaria, vol. 20, n° 5, 2019.
- [15] A. José, «EDteam,» 01 12 2017. [En línea]. Available: <https://ed.team/blog/renderiza-componentes-de-react-en-distintos-elementos-del-dom>. [Último acceso: 20 11 2021].
- [16] L. Valdez y C. Tejada, «Alltime IoT: Tecnología para la Educación,» Bogotá, Colombia, 2018.
- [17] «todo-redes,» 2020. [En línea]. Available: <https://todo-redes.com/equipos-de-redes/gateway-puerta-de-enlace>.
- [18] P. P. Ray, “A survey of IoT cloud platforms,” Futur. Comput. Informatics J., vol. 1, no. 1–2, pp. 35–46, 2016, doi: 10.1016/j.fcij.2017.02.001.
- [19] L. Ordoñez, «By Orange,» 22 7 2020. [En línea]. Available: <https://blog.orange.es/red/que-es-bluetooth/>. [Último acceso: 10 10 2021].
- [20] sd R. Hat, «Red Hat,» 07 03 2018. [En línea]. Available: <https://www.redhat.com/es/topics/cloud-computing/what-is-cloud-anagement#:~:text=La%20gesti%C3%B3n%20de%20la%20nube%20es%20la%20forma%20en%20que,las%20aplicaciones%20y%20los%20servicios>. [Último acceso: 03 01 2022].
- [21] J. Martínez, «IrDa definicion,» 10 12 2019. [En línea]. Available: <http://www.dit.upm.es/~jantonio/documentos/revistas/irda/irda2.html>. [Último acceso: 10 10 2021].
- [22] J. Moreno, «Informe Técnico: Protocolo ZigBee (IEEE 802.15.4),» 1 06 2007. [En línea]. Available: https://rua.ua.es/dspace/bitstream/10045/1109/7/Informe_ZigBee.pdf. [Último acceso: 10 10 2021].

- [23] J. Penalva, «NFC,» 10 10 2019. [En línea]. Available: <https://www.xataka.com/moviles/nfc-que-es-y-para-que-sirve>. [Último acceso: 11 10 2021].
- [24] E. Alvarez, «Computer Hoy,» 27 03 2018. [En línea]. Available: <https://computerhoy.com/noticias/internet/diccionario-del-wifi-80211-bgn-banda-dual-5-ghz-78159>. [Último acceso: 11 10 2021].
- [25] L. Dra. González, «Clínica Veterinaria RIOSECO,» 25 Febrero 2021. [En línea]. Available: <https://riosecoclinicaveterinaria.es/cardiopatias-en-perros/>. [Último acceso: 29 10 2021].
- [26] J. C. J. V. Lizárraga, «“Evaluación Comparativa de la Saturación de Oxígeno en Caninos”,» Universidad Católica de Santa María, Arequipa, Perú, 2017.
- [27] PetPace, «PetPace,» PetPace, 12 01 2019. [En línea]. Available: <http://petpace.es/>. [Último acceso: 29 10 2021].
- [28] MemoPet, «MemoPet,» MemoPet, 15 05 2020. [En línea]. Available: <https://www.piensoadomicilio.es/inteligentes/5464-collar-inteligente-memopet-rojo.html>. [Último acceso: 29 10 2021].
- [29] L. Ruiz, «Mente,» 14 10 2018. [En línea]. Available: <https://psicologiaymente.com/miscelanea/alfa-de-cronbach>. [Último acceso: 11 10 2021].
- [30] D. Arduino, «DescubreArduino.com,» 01 05 2021. [En línea]. Available: <https://descubrearduino.com/comparamos-varias-de-las-arduino-mas-utilizadas/>. [Último acceso: 05 01 2022].
- [31] S. Hector y M. Mayra, «Oximetría de pulso,» Scielo, vol. 51, nº 2, pp. 05-10, 2012.
- [32] C. María, «Diario Electrónico Hoy,» 11 05 2015. [En línea]. Available: <https://www.diarioelectronicohoy.com/oximetro-y-sensor-de-ritmo-cardiaco/#:~:text=El%20MAX30100%20es%20un%20circuito,de%20ox%C3%ADgeno%20en%20la%20sangre..> [Último acceso: 10 01 2022].
- [33] C. Ruth, «Diseño e Implementación de una Aplicación Móvil Basada en Android para el Evaluación de Anemia Ferropénica,» Cuenca, 2018.
- [34] mechatronics.com, «mechatronics.com,» mechatronics.com, 01 12 2020. [En línea]. Available: <https://naylampmechatronics.com/oled/391-display-oled-i2c-091-12832-ssd1306.html>. [Último acceso: 15 01 2022].

Anexo B: Hoja de datos Sensor MAX 30100

En el anexo B se observa la hoja de datos técnicos del sensor MAX30100 con cada una de sus características eléctricas, funcionales y distribución de pines.



Pin Description

| PIN | NAME | FUNCTION |
|-------------|-----------------|--|
| 1, 7, 8, 14 | N.C. | No Connection. Connect to PCB Pad for Mechanical Stability. |
| 2 | SCL | I ² C Clock Input |
| 3 | SDA | I ² C Clock Data, Bidirectional (Open-Drain) |
| 4 | PGND | Power Ground of the LED Driver Blocks |
| 5 | IR_DRV | IR LED Cathode and LED Driver Connection Point. Leave floating in circuit. |
| 6 | R_DRV | Red LED Cathode and LED Driver Connection Point. Leave floating in circuit. |
| 9 | R_LED+ | Power Supply (Anode Connection) for Red LED. Bypass to PGND for best performance. Connected to IR_LED+ internally. |
| 10 | IR_LED+ | Power Supply (Anode Connection) for IR LED. Bypass to PGND for best performance. Connected to R_LED+ internally. |
| 11 | V _{DD} | Analog Power Supply Input. Bypass to GND for best performance. |
| 12 | GND | Analog Ground |
| 13 | INT | Active-Low Interrupt (Open-Drain) |

Electrical Characteristics

(V_{DD} = 1.8V, V_{IR_LED+} = V_{R_LED+} = 3.3V, T_A = +25°C, min/max are from T_A = -40°C to +85°C, unless otherwise noted.) (Note 2)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|--|-------------------|--|-----|-----|------|-------|
| POWER SUPPLY | | | | | | |
| Power-Supply Voltage | V _{DD} | Guaranteed by RED and IR count tolerance | 1.7 | 1.8 | 2.0 | V |
| LED Supply Voltage (R_LED+ or IR_LED+ to PGND) | V _{LED+} | Guaranteed by PSRR of LED Driver | 3.1 | 3.3 | 5.0 | V |
| Supply Current | I _{DD} | SpO ₂ and heart rate modes, PW = 200μs, 50sps | | 600 | 1200 | μA |
| | | Heart rate only mode, PW = 200μs, 50sps | | 600 | 1200 | |
| Supply Current in Shutdown | I _{SHDN} | T _A = +25°C, MODE = 0x80 | | 0.7 | 10 | μA |

Anexo C: Hoja de datos – Pantalla OLED

En el anexo C se observa de igual manera la hoja de datos técnicos de la pantalla OLED, dispositivo utilizado en la implementación del prototipo, con sus características mecánicas, eléctricas y rangos de funcionamiento.



FEATURES

- Type: graphic
- Display format: 128 x 32 dots
- Built-in controller: SSD1306BZ
- Duty cycle: 1/32
- +3 V power supply
- Interface: 6800
- With polarizer
- Material categorization: for definitions of compliance please see www.vishay.com/doc?99912



| MECHANICAL DATA | | |
|------------------|--------------------|------|
| ITEM | STANDARD VALUE | UNIT |
| Module dimension | 30.0 x 11.5 x 1.45 | mm |
| Viewing area | 24.384 x 7.584 | |
| Active area | 22.384 x 5.584 | |
| Dot size | 0.152 x 0.152 | |
| Dot pitch | 0.175 x 0.175 | |
| Mounting hole | n/a | |

| ABSOLUTE MAXIMUM RATINGS | | | | |
|--|------------------|----------------|------|------|
| ITEM | SYMBOL | STANDARD VALUE | | UNIT |
| | | MIN. | MAX. | |
| Supply voltage for logic ⁽¹⁾⁽²⁾ | V _{DD} | 0 | 4 | V |
| Supply voltage for display ⁽¹⁾⁽²⁾ | V _{CC} | 0 | 16 | V |
| Operating temperature | T _{OP} | -40 | +80 | °C |
| Storage temperature | T _{STG} | -40 | +80 | |

| ELECTRICAL CHARACTERISTICS | | | | | | |
|---|-----------------|--------------------------|---------------------|------|---------------------|------|
| ITEM | SYMBOL | CONDITION | STANDARD VALUE | | | UNIT |
| | | | MIN. | TYP. | MAX. | |
| Supply voltage for logic | V _{DD} | - | 2.8 | 3.0 | 3.3 | V |
| Supply voltage for display | V _{CC} | - | 7.0 | 7.25 | 8.0 | |
| Input high voltage | V _{IH} | - | 0.8 V _{DD} | - | V _{DDI/O} | |
| Input low voltage | V _{IL} | - | 0 | - | 0.2 V _{DD} | |
| Output high voltage | V _{OH} | - | 0.9 V _{DD} | - | V _{DDI/O} | |
| Output low voltage | V _{OL} | - | 0 | - | 0.1 V _{DD} | |
| Operating current for V _{CC} (V _{CC} supplied externally) | I _{CC} | V _{CC} = 7.25 V | 5 | 6 | 7 | mA |

| | | | | | CAN NENA | | | | | | | | | | | |
|-------------------|---------------------|---------------------|---------------------|--------------------|--|-------------|----------------------|----------|---|----|------------|------|-------|-------|----------------------|--|
| | | | | | | | | | | | | | | | | |
| Número Mediciones | Pechera BPMs Dia 1 | Pechera BPMs Dia 2 | Pechera BPMs Dia 3 | Pechera BPMs Suma | Ecuación a desarrollar | | Rangos de Intervalos | | | | | | | | | |
| 1 | 84 | 83 | 85 | 252 | $\alpha = \frac{K}{K-1} \left(1 - \frac{\sum V_i}{V_t} \right)$ <table border="1"> <thead> <tr> <th colspan="2">Datos</th> </tr> </thead> <tbody> <tr> <td>K</td> <td>30</td> </tr> <tr> <td>$\sum V_i$</td> <td>6,23</td> </tr> <tr> <td>V_t</td> <td>11,05</td> </tr> </tbody> </table> | | Datos | | K | 30 | $\sum V_i$ | 6,23 | V_t | 11,05 | Rangos de Intervalos | |
| Datos | | | | | | | | | | | | | | | | |
| K | 30 | | | | | | | | | | | | | | | |
| $\sum V_i$ | 6,23 | | | | | | | | | | | | | | | |
| V_t | 11,05 | | | | | | | | | | | | | | | |
| 2 | 83 | 84 | 86 | 253 | | | 0-0,2 | Muy Baja | | | | | | | | |
| 3 | 84 | 85 | 84 | 253 | | | 0,2-0,4 | Baja | | | | | | | | |
| 4 | 80 | 84 | 83 | 247 | | | 0,4-0,6 | Moderada | | | | | | | | |
| 5 | 80 | 83 | 83 | 246 | | | 0,6-0,8 | Buena | | | | | | | | |
| 6 | 82 | 84 | 82 | 248 | | | 0,8-1 | Alta | | | | | | | | |
| 7 | 83 | 84 | 83 | 250 | | | | | | | | | | | | |
| 8 | 86 | 86 | 85 | 257 | | | | | | | | | | | | |
| 9 | 80 | 83 | 84 | 247 | | | | | | | | | | | | |
| 10 | 83 | 83 | 86 | 252 | | | | | | | | | | | | |
| Promedio | 82,5 | 83,9 | 84,1 | 250,5 | | | | | | | | | | | | |
| Varianza | 3,65 | 0,89 | 1,69 | 11,05 | Alfa Cronbach | 0,451240443 | | | | | | | | | | |
| Desviación | 1,910497317 | 0,943398113 | 1,3 | 3,324154028 | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | |
| Número Mediciones | Pechera SPO2% Dia 1 | Pechera SPO2% Dia 2 | Pechera SPO2% Dia 3 | Pechera SPO2% Suma | Ecuación a desarrollar | | Rangos de Intervalos | | | | | | | | | |
| 1 | 99 | 96 | 96 | 291 | $\alpha = \frac{K}{K-1} \left(1 - \frac{\sum V_i}{V_t} \right)$ <table border="1"> <thead> <tr> <th colspan="2">Datos</th> </tr> </thead> <tbody> <tr> <td>K</td> <td>30</td> </tr> <tr> <td>$\sum V_i$</td> <td>1,9</td> </tr> <tr> <td>V_t</td> <td>3,89</td> </tr> </tbody> </table> | | Datos | | K | 30 | $\sum V_i$ | 1,9 | V_t | 3,89 | Rangos de Intervalos | |
| Datos | | | | | | | | | | | | | | | | |
| K | 30 | | | | | | | | | | | | | | | |
| $\sum V_i$ | 1,9 | | | | | | | | | | | | | | | |
| V_t | 3,89 | | | | | | | | | | | | | | | |
| 2 | 98 | 95 | 97 | 290 | | | 0-0,2 | Muy Baja | | | | | | | | |
| 3 | 98 | 96 | 97 | 291 | | | 0,2-0,4 | Baja | | | | | | | | |
| 4 | 99 | 96 | 98 | 293 | | | 0,4-0,6 | Moderada | | | | | | | | |
| 5 | 99 | 96 | 98 | 293 | | | 0,6-0,8 | Buena | | | | | | | | |
| 6 | 99 | 97 | 97 | 293 | | | 0,8-1 | Alta | | | | | | | | |
| 7 | 98 | 97 | 98 | 293 | | | | | | | | | | | | |
| 8 | 98 | 98 | 99 | 295 | | | | | | | | | | | | |
| 9 | 99 | 98 | 99 | 296 | | | | | | | | | | | | |
| 10 | 99 | 98 | 99 | 296 | | | | | | | | | | | | |
| Promedio | 98,6 | 96,7 | 97,8 | 293,1 | | | | | | | | | | | | |
| Varianza | 0,24 | 0,7 | 0,96 | 3,89 | Alfa Cronbach | 0,529208404 | | | | | | | | | | |
| Desviación | 0,516397779 | 1,059349905 | 1,032795559 | 2,078995484 | | | | | | | | | | | | |

| CAN OJITO | | | | |
|-------------------|--------------------|--------------------|--------------------|-------------------|
| Número Mediciones | Pechera BPMs Día 1 | Pechera BPMs Día 2 | Pechera BPMs Día 3 | Pechera BPMs Suma |
| 1 | 132 | 136 | 144 | 412 |
| 2 | 136 | 136 | 145 | 417 |
| 3 | 145 | 138 | 145 | 428 |
| 4 | 147 | 140 | 143 | 430 |
| 5 | 150 | 145 | 143 | 438 |
| 6 | 155 | 148 | 143 | 446 |
| 7 | 155 | 148 | 142 | 445 |
| 8 | 155 | 150 | 142 | 447 |
| 9 | 155 | 151 | 143 | 449 |
| 10 | 155 | 151 | 143 | 449 |
| Promedio | 148,50 | 144,30 | 143,30 | 436,10 |
| Varianza | 72,94 | 38,46 | 1,12 | 186,77 |
| Desviación | 8,54 | 6,20 | 1,06 | 13,67 |

| Ecuación a desarrollar | | Rangos de Intervalos | |
|--|-------------|----------------------|----------|
| $\alpha = \frac{K}{K-1} \left(1 - \frac{\sum V_i}{V_t} \right)$ | | 0-0,2 | Muy Baja |
| | | 0,2-0,4 | Baja |
| | | 0,4-0,6 | Moderada |
| | | 0,6-0,8 | Buena |
| Datos | | 0,8-1 | Alta |
| K | 3 | | |
| $\sum V_i$ | 112,5222222 | | |
| V_t | 186,7666667 | | |

| | |
|---------------------|-------------|
| Alfa Cronbad | 0,596287703 |
|---------------------|-------------|

| Número Mediciones | Pechera SPO2% Día 1 | Pechera SPO2% Día 2 | Pechera SPO2% Día 3 | Pechera SPO2% Suma |
|-------------------|---------------------|---------------------|---------------------|--------------------|
| 1 | 94 | 96 | 95 | 285 |
| 2 | 94 | 95 | 95 | 284 |
| 3 | 94 | 94 | 94 | 282 |
| 4 | 94 | 94 | 94 | 282 |
| 5 | 94 | 96 | 94 | 284 |
| 6 | 95 | 95 | 95 | 285 |
| 7 | 94 | 94 | 94 | 282 |
| 8 | 94 | 96 | 96 | 286 |
| 9 | 94 | 96 | 96 | 286 |
| 10 | 93 | 96 | 95 | 284 |
| Promedio | 94 | 95,2 | 94,8 | 284 |
| Varianza | 0,2 | 0,7 | 0,56 | 2,2 |
| Desviación | 0,471404521 | 0,918936583 | 0,788810638 | 1,56347192 |

| Ecuación a desarrollar | | Rangos de Intervalos | |
|--|------|----------------------|----------|
| $\alpha = \frac{K}{K-1} \left(1 - \frac{\sum V_i}{V_t} \right)$ | | 0-0,2 | Muy Baja |
| | | 0,2-0,4 | Baja |
| | | 0,4-0,6 | Moderada |
| | | 0,6-0,8 | Buena |
| Datos | | 0,8-1 | Alta |
| K | 3 | | |
| $\sum V_i$ | 1,46 | | |
| V_t | 2,2 | | |

| | |
|---------------------|-------------|
| Alfa Cronbad | 0,504545455 |
|---------------------|-------------|

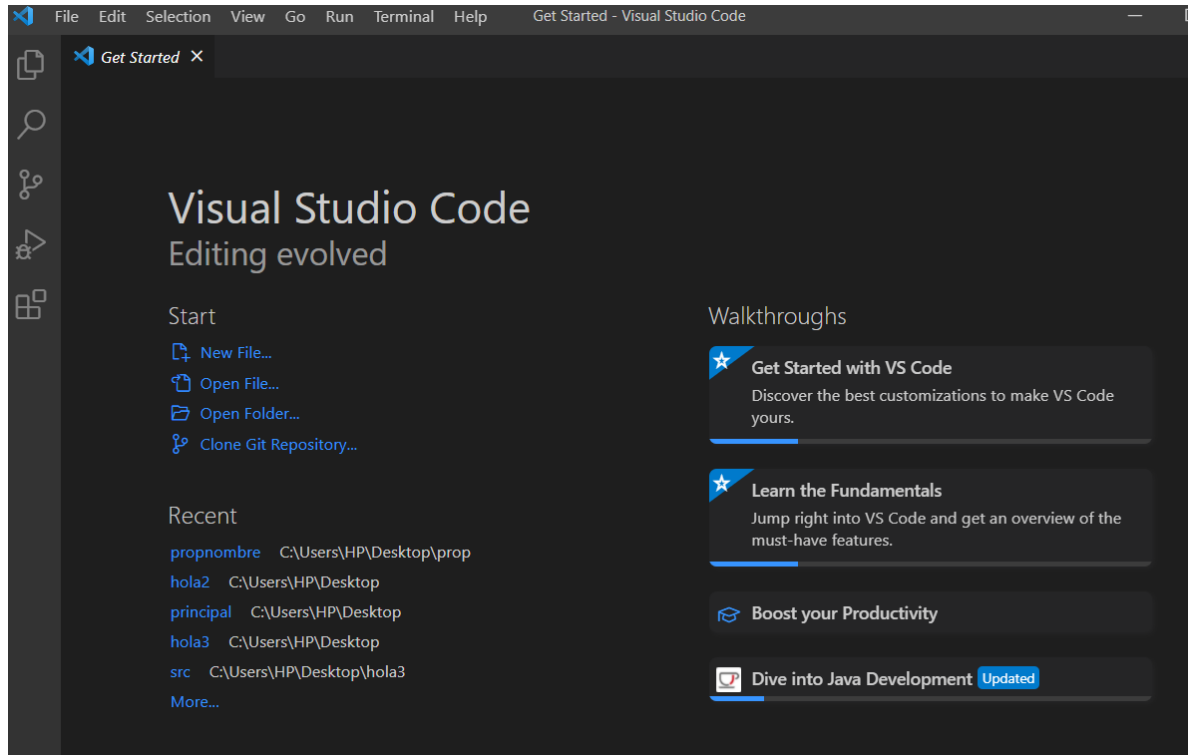
| CAN KUKY HOLTER | | | | |
|-------------------|-------------------|-------------------|-------------------|-----------------------|
| Número Mediciones | Instrumento Día 1 | Instrumento Día 2 | Instrumento Día 3 | Instrumento BPMs Suma |
| 1 | 136 | 135 | 137 | 408 |
| 2 | 136 | 135 | 137 | 408 |
| 3 | 135 | 135 | 137 | 407 |
| 4 | 135 | 135 | 136 | 406 |
| 5 | 134 | 134 | 136 | 404 |
| 6 | 134 | 135 | 135 | 404 |
| 7 | 134 | 134 | 135 | 403 |
| 8 | 135 | 134 | 135 | 404 |
| 9 | 134 | 134 | 136 | 404 |
| 10 | 134 | 134 | 135 | 403 |
| Promedio | 134,7 | 134,5 | 135,9 | 405,1 |
| Varianza | 0,677777778 | 0,277777778 | 0,766666667 | 3,877777778 |
| Desviación | 0,823272602 | 0,527046277 | 0,875595036 | 1,969207398 |

| Ecuación a desarrollar | | Rangos de Intervalos | |
|--|-------------|----------------------|----------|
| $\alpha = \frac{K}{K-1} \left(1 - \frac{\sum V_i}{V_t} \right)$ | | 0-0,2 | Muy Baja |
| | | 0,2-0,4 | Baja |
| | | 0,4-0,6 | Moderada |
| | | 0,6-0,8 | Buena |
| Datos | | 0,8-1 | Alta |
| K | 3 | | |
| $\sum V_i$ | 1,722222222 | | |
| V_t | 3,877777778 | | |

| | |
|---------------------|-------------|
| Alfa Cronbad | 0,833810888 |
|---------------------|-------------|

Anexo F: Software Visual Studio Code.

En el anexo F se visualiza la pantalla de inicio de Visual Studio Code, entorno de desarrollo a utilizar para la programación de la aplicación móvil en React Native.



Anexo G: Código Pechera en Arduino IDE.

En el anexo G se puede observar el código de programación realizado para poder implementar el circuito diseñado, este código fue compilado y ejecutado dentro de Arduino IDE para posteriormente ser subido a la placa Arduino Pro Mini.

```
#include <Wire.h>//Librería para comunicación I2C

#include "MAX30100_PulseOximeter.h">//Librería para reconocimiento de sensor
MAX30100 como extension pulsioxímetro

#include "MAX30100.h">//Librería para reconocimiento de sensor MAX30100

#include <U8g2lib.h>//Librería para habilitar pantalla OLED

#define REPORTING_PERIOD_MS    1500

#define PULSE_WIDTH            MAX30100_SPC_PW_1600US_16BITS

#define IR_LED_CURRENT         MAX30100_LED_CURR_40MA

#define LED_CURRENT            MAX30100_LED_CURR_20_8MA

#define SAMPLING_RATE          MAX30100_SAMPRATE_100HZ

U8G2_SSD1306_128X32_UNIVISION_F_HW_I2C u8g2(U8G2_R0);

PulseOximeter pox;

MAX30100 sensor;

uint32_t tsLastReport = 0;

uint32_t last_beat=0;

bool initialized=false;

int HRclean;

int SpO2;
```



```
float lectura;  
  
float volt;  
  
byte ledR=2; // LED rojo  
  
byte ledA=3; // LED azul  
  
byte ledV=4; // LED verde
```

```
void onBeatDetected()  
  
{  
  
  show_beat();  
  
  last_beat=millis();  
  
}
```

```
void show_beat()  
  
{  
  
  u8g2.setFont(u8g2_font_cursor_tr);  
  
  u8g2.setCursor(118,10);  
  
  u8g2.print("_");  
  
  u8g2.sendBuffer();  
  
}
```

```
void initial_display()  
  
{  
  
  if (not initialized)
```

```

{
    u8g2.clearBuffer();

    u8g2.setCursor(15,12);

    u8g2.setFont(u8g2_font_crox2hb_tr);

    u8g2.print("CONTROL PET");

    u8g2.setFont(u8g2_font_crox2h_tr);

    u8g2.setCursor(30,29);

    u8g2.print("Iniciando...");

    u8g2.sendBuffer();

    delay(4000);

    initialized=true;

    u8g2.clearBuffer();

    u8g2.setFont(u8g2_font_crox2hb_tr);

    if (!pox.begin()) {

        u8g2.setCursor(40,12);

        u8g2.print("FAILED");

        u8g2.setCursor(15,29);

        u8g2.print("Check Sensor !");

        u8g2.sendBuffer();

        for(;;);
    }
}

```

```

    } else {
        u8g2.setCursor(20,12);
        u8g2.print("Bienvenido");
        u8g2.setCursor(0,29);
        u8g2.print("Coloque Pechera");
        u8g2.sendBuffer();
    }
    delay(2000);
}
}

void setup()
{
    Serial.begin(9600);
    u8g2.begin();
    initial_display();
    pox.begin();
    pox.setOnBeatDetectedCallback(onBeatDetected);
    pox.setIRLedCurrent(LED_CURRENT);
    sensor.setMode(MAX30100_MODE_SPO2_HR);
    sensor.setLedsPulseWidth(PULSE_WIDTH);
    sensor.setSamplingRate(SAMPLING_RATE);
    pinMode(A0,INPUT);
}

```

```
pinMode(ledR, OUTPUT);  
pinMode(ledA, OUTPUT);  
pinMode(ledV, OUTPUT);  
digitalWrite(ledR, LOW);  
digitalWrite(ledA, LOW);  
digitalWrite(ledV, LOW);  
}  
void loop()  
{  
  lectura = analogRead(A0);  
  volt = lectura /1023 * 5.0;  
  if (volt<=3.1){  
    digitalWrite(ledR, LOW);  
    digitalWrite(ledA, HIGH);  
    digitalWrite(ledV, HIGH);  
  }  
  else if(volt>=3.5 && volt<=3.6){  
    digitalWrite(ledR, HIGH);  
    digitalWrite(ledA, LOW);  
    digitalWrite(ledV, HIGH);  
  }  
  else if(volt>=3.9){
```

```

digitalWrite(ledR, HIGH);

digitalWrite(ledA, HIGH);

digitalWrite(ledV, LOW);

}

pox.update();

HRclean = pox.getHeartRate();

SpO2 = pox.getSpO2();

if ((millis() - tsLastReport > REPORTING_PERIOD_MS)) {

    u8g2.clearBuffer();

    u8g2.setFont(u8g2_font_crox2h_tr);

    u8g2.setCursor(0,12);

    u8g2.print("HR");

    u8g2.setCursor(75,12);

    u8g2.print("Bpm");

    u8g2.setCursor(0,30);

    u8g2.print("SpO2 ");

    u8g2.setCursor(75,30);

    u8g2.print("%");

    u8g2.setFont(u8g2_font_fub11_tf);

    u8g2.setCursor(45,12);

    u8g2.print(HRclean);

    u8g2.setCursor(45,30);

```

```
u8g2.print(SpO2);  
  
u8g2.setFont(u8g2_font_cursor_tr);  
  
u8g2.setCursor(118,10);  
  
u8g2.print("^");  
  
u8g2.sendBuffer();  
  
Serial.print(HRclean);  
  
Serial.print(',');  
  
Serial.print(SpO2);  
  
Serial.print(',');  
  
tsLastReport = millis();  
  
}  
  
}
```

Anexo H: Código Aplicación móvil en React Native.

En el anexo H se visualiza cada archivo .js que componen la aplicación móvil desarrollada en React Native.

Splash.js

```
import React, {Component, component} from "react";

import { View, Text } from "react-native";

import LottieView from 'lottie-react-native';

export default class Splash extends Component {

  constructor(props) {

    super();

  }

  render() {

    return(

      <View

        style={{ flex:1,

          backgroundColor:'#ffffff'

        }}

      >

      <LottieView source={require('../assets/dogsplash.json')}

        autoPlay

        loop={false}

        speed= {1.2}
```

```

        onAnimationFinish={()=>{
            console.log('ter')
            this.props.navigation.replace('Profile');
        }}
    />
</View>
)
}
}

```

Profile.js

```

import React, {Component} from "react";
import { View, Text, StyleSheet,TouchableOpacity, Button} from "react-native";
import LottieView from 'lottie-react-native';
export default class Profile extends Component {
    constructor(props) {
        super();
    }
    render() {
        return(
            <View style={styles.container}>
                <LottieView source={require('../assets/love.json')}
                    autoPlay

```



```

loop={false}
speed= {1.2} />
<Text style={styles.header}>BIENVENIDO</Text>
<Text style={styles.header2}>Identifíquese</Text>
  <View style={styles.container2}>
    <TouchableOpacity
      style={styles.btn}
      onPress={this.login2} >
      <Text style={styles.text}>MÉDICO</Text>
    </TouchableOpacity>
    <TouchableOpacity
      style={styles.btn2}
      onPress={this.login3} >
      <Text style={styles.text}>CLIENTE</Text>
    </TouchableOpacity>
  </View>
</View>
)
}
login2 = ()=> {
  this.props.navigation.navigate('Home');
}

```

```

login3 = ()=> {
    this.props.navigation.navigate('HomeCliente');
}
}

const styles=StyleSheet.create({
    container:{
        flex:1,
        alignItems:'center',
        justifyContent:'center',
        backgroundColor:'#F9AD84' },
    container2:{
        flexDirection:'row',
        alignItems:'center',
        justifyContent: 'space-between',
        marginBottom: 10,    },
    text:{
        color:'#fff',
        fontSize:10
    }, btn:{
        flexDirection: 'row',
        backgroundColor:'#42C1E5',
        alignItems:'center',

```

```
borderRadius:30,
height:38,
width:68,
justifyContent:'center',
marginRight:15,  },
btn2:{
  flexDirection: 'row',
  backgroundColor:'#42C1E5',
  alignItems:'center',
  borderRadius:20,
  justifyContent:'center',
  height:38,
  width:68,
},
header:{
  fontSize:24,
  marginBottom:310,
  color:'#156EB1',
  fontWeight:'bold',
  fontFamily: 'Metric-Regular',
  fontStyle: 'italic',
},
```

```
header2:{
  fontSize:18,
  marginBottom:0,
  color:'#DC5618',
  fontWeight:'bold',
  fontFamily: 'Metric-Regular',
}
})
```

Logeo de sesión con Express

```
var express = require('express');
var router = express.Router();
var mysql = require('mysql');
var connection = mysql.createConnection({
  host:'18.188.51.63',
  port:'3306',
  user:'root',
  password:'yuli123456',
  database:'Almacenamiento'
});
router.post('/', function(req, res, next) {
  var nombre = req.body.nombre;
```

```

var apellido = req.body.apellido;

var codigom = req.body.codigom;

connection.query(

    "SELECT * FROM medicos WHERE nombre = ? AND apellido = ? AND
codigom = ?",

    [nombre, apellido, codigom], function (err, row, field) {

        if (err) {

            console.log(err);

            res.send({ 'success': false, 'message': 'No se puede conectar bd' });

        }

        if (row.length > 0) {

            res.send({ 'success': true, 'user': row[0].apellido });

        } else {

            res.send({ 'success': false, 'message': 'Usuario incorrecto' });

        }

    });

});

module.exports = router;

```

Anexo I: Códigos para activación central Asterisk

Sip.conf

[general]

context=internal

allowguest=no
allowoverlap=no
bindport=5060
bindaddr=0.0.0.0
srvlookup=no
disallow=all
allow=ulaw
alwaysauthreject=yes
canreinvite=no
nat=yes
session-timers=refuse
localnet=192.168.1.0/255.255.255.0
externip=3.140.208.49
[7001]
type=friend
host=dynamic
secret=123
context=internal
[7002]
type=friend
host=dynamic
secret=456

context=internal

Extensions.conf

[internal]

exten => 5,1,Goto(consulta,s,1)

exten => 7001,1,Dial(PJSIP/7001,20,Ttm)

same => n,Hangup

exten => 99,1,Answer(500)

same => n,Record(en/alerta.gsm)

same => n,Wait(1)

same => n,Playback(alerta)

same => n,Hangup()

exten => 105,1,Noop(-----Demo tts-----)

exten=>105,2,agi(/var/lib/asterisk/agi-bin/asterisk-googletts/googletts.agi,"Bienvenido al sistema virtual de Veterinaria Mascotas Felices.",es)

exten => 105,3,agi(/var/lib/asterisk/agi-bin/asterisk-googletts/googletts.agi,"Presione 1 para consultoría de vacunas.",es)

exten => 105,4,agi(/var/lib/asterisk/agi-bin/asterisk-googletts/googletts.agi,"Presione 2 para consultoría de signos vitales.",es)

```
exten => 105,5,agi(/var/lib/asterisk/agi-bin/asterisk-  
googletts/googletts.agi,"Presione 3 para llamar a atención al cliente.",es)
```

```
exten => 105,6,WaitExten()
```

```
exten => 1,1,goto(IVR1,s,1)
```

```
exten => 2,1,goto(IVR2,s,1)
```

```
exten => 3,1,goto(ASISTENTE,7001,1)
```

```
exten => i,1,agi(/var/lib/asterisk/agi-bin/asterisk-googletts/googletts.agi,"Código  
inválido, adios.",es)
```

```
exten => i,2,Hangup
```

```
[IVR1]
```

```
exten => s,1,Answer()
```

```
exten => s,2,AGI(vacuna.php)
```

```
exten => s,3,Hangup
```

```
[IVR2]
```

```
exten => s,1,Answer()
```

```
exten => s,2,AGI(leer.php)
```

```
exten => s,3,Hangup
```

```
[ASISTENTE]
```

```
exten => 7001,1,Dial(SIP/7001)
```

```
Leer.php
```

```
#!/usr/bin/php
```

```
<?php
```



```

require('phpagi.php');

error_reporting(E_ALL);

$agi = new AGI();

$agi->answer();

$agi->exec("AGI", "google tts.agi, \"Ingrese el código del paciente.\", es");

$result = $agi->get_data('beep', 3000, 20);

$codigo = $result['result'];

$agi->verbose("Conectado...".$callerid);

$connection = new mysqli('localhost', 'root', 'yuli123456', 'Almacenamiento');

$sql = 'SELECT * FROM medicion where id='.$codigo;

$sql2 = 'SELECT username FROM user where id='.$codigo;

$result = $connection->query( $sql);

$result2 = $connection->query( $sql2);

echo $result;

echo $result2;

$_re=$result->fetch_array(MYSQLI_ASSOC);

$_re2=$result2->fetch_array(MYSQLI_ASSOC);

$agi->exec("AGI", "google tts.agi, \"La última medición de la pechera de
$_re2[username] fue $_re[HR] en frecuencia cardiaca y $_re[SPO2] en saturación
de oxígeno en la sangr$

$agi->hangup();

```

Vacuna.php

```

#!/usr/bin/php

<?php

require('phpagi.php');

error_reporting(E_ALL);

$agi = new AGI();

$agi->answer();

$agi->exec("AGI","googletts.agi,\"Ingrese el código del paciente.\",es");

$result = $agi->get_data('beep',3000,20);

$codigo = $result['result'];

$agi->verbose("Conectado...".$callerid);

$connection = new mysqli('localhost', 'root', 'yuli123456','Almacenamiento');

$sql = 'SELECT * FROM user where id='.$codigo;

$result = $connection->query( $sql);

echo $result;

$_re=$result->fetch_array(MYSQLI_ASSOC);

    if($_re['vacuna5']==null){

        if($_re['vacuna4']==null){

            if($_re['vacuna3']==null){

                if($_re['vacuna2']==null){

                    $agi->exec("AGI","googletts.agi,\"La última vacuna de
$_re[username] fue $_re[vacuna1] el $_re[fecha1] y su próxima vacuna es $_re[$

                }
            }
        }
    }

```

```

        else{

                $agi->exec("AGI","googlets.agi,\"La última vacuna
de $_re[username] fue $_re[vacuna2] el $_re[fecha2] y su próxima vacuna $

                }

        }

        else{

                $agi->exec("AGI","googlets.agi,\"La última vacuna de
$_re[username] fue $_re[vacuna3] el $_re[fecha3] y su próxima vacuna es $_re$

                }

        }

        else{

                $agi->exec("AGI","googlets.agi,\"La última vacuna de
$_re[username] fue $_re[vacuna4] el $_re[fecha4] y su próxima vacuna es
$_re[fecha5].$

                }

        }

        else{

                $agi->exec("AGI","googlets.agi,\"La última vacuna de $_re[username]
fue $_re[vacuna5] el $_re[fecha5].\",es");

                }

        $agi->hangup();

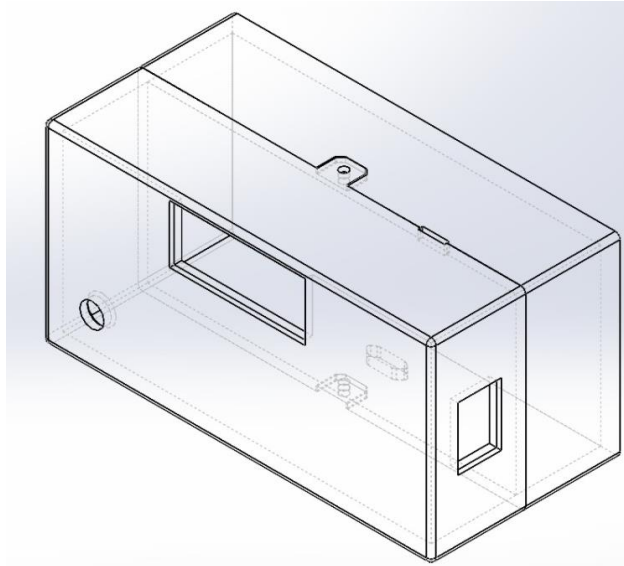
```

Anexo J: Realización del case del dispositivo de monitoreo de mascotas.

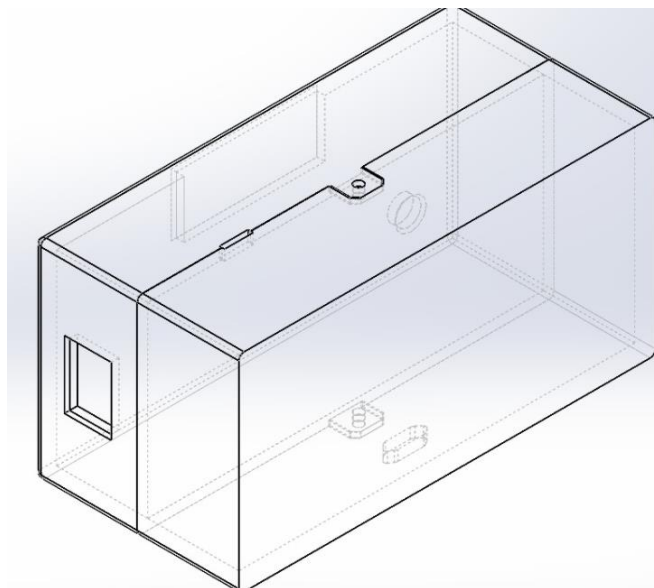
En el anexo J se visualiza las etapas de diseño del case de protección que tendrá nuestro circuito impreso para posteriormente ensamblarlo en la pechera.

Diseño del case

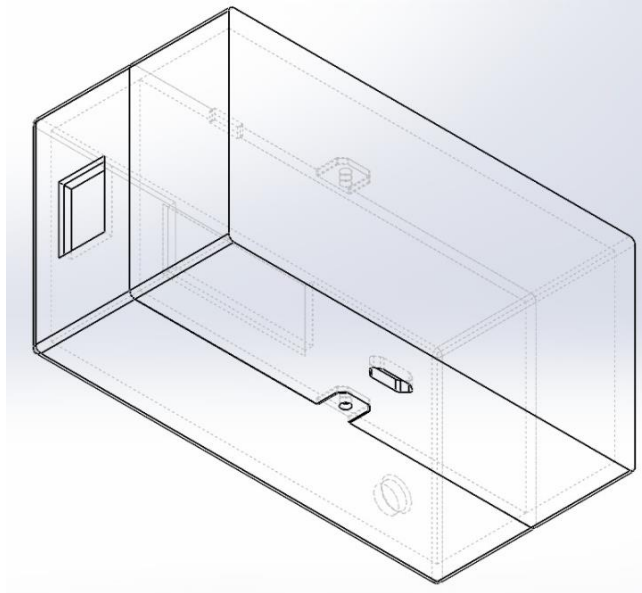
Parte Frontal



Parte Lateral y trasera



Parte Inferior



Impresión 3D



Case ensamblado con circuito



Anexo K: Pruebas de funcionamiento

En el anexo K se visualiza las pruebas realizadas en los canes de estudio, el monitoreo dentro de la aplicación móvil y el asistente de consultoría VoIP con el uso de Zoiper que es el softphone o software más utilizado al momento de realizar llamadas de este tipo a través del Internet.

