



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

**Diseño y creación de un sistema de información para
almacenamiento, visualización y alertas de signos
vitales**

SERGIO ANDRÉS LAGOS FUQUEN
MATEO PARRA POSSE

UNIVERSIDAD SANTO TOMÁS
INGENIERÍA ELECTRÓNICA
BOGOTÁ D.C

2022

**Diseño y creación de un sistema de información para
almacenamiento, visualización y alertas de signos
vitales**

SERGIO ANDRÉS LAGOS FUQUEN
MATEO PARRA POSSE

Proyecto de grado presentado como requisito para optar al título
de INGENIERO ELECTRÓNICO

UNIVERSIDAD SANTO TOMÁS
INGENIERÍA ELECTRÓNICA
BOGOTÁ D.C
2022

UNIVERSIDAD SANTO TOMÁS

Ingeniería Electrónica

Este proyecto de grado fue aceptado para la carrera de Ingeniero(a)
Electrónico(a) en Junio 2019

Director:

Ing. Dario Segura Torres

Jurados:

Ing. Gianina Garrido Silva

Ing. Edwin Francisco Forero García

Sustentación proyecto de grado: 25 mayo 2022, BOGOTÁ D.C

Sergio Andrés Lagos Fuquen

Mateo Parra Posse



UNIVERSIDAD SANTO TOMÁS



FACULTAD DE INGENIERÍA ELECTRÓNICA

UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA
TESIS DE GRADO

El trabajo de grado titulado «**Diseño y creación de un sistema de información para almacenamiento, visualización y alertas de signos vitales**» realizado por los estudiantes **Sergio Andrés Lagos Fuquen** y **Mateo Parra Posse**, con código estudiantil 2200219 y 2199429 cumplen con los requisitos exigidos por la **Universidad Santo Tomás** para optar al título de **Ingeniero Electrónico**.

Ing. Dario Segura Torres

Ing. Gianina Garrido Silva

Ing. Edwin Francisco Forero García

DEDICATORIA

*Dedicado a mi mamá Liliana Posse Rodríguez
a mi tío Álvaro Posse Rodríguez
a mi novia Alejandra Vargas Guerrero y a mis amigos
por todo el apoyo y amor brindado todo este tiempo
Mateo Parra Posse.*

*Dedicado a mi familia y pareja por darme el apoyo incondicional
siempre dándome fuerzas para seguir;
a mis amigos por sus buenos ánimos y deseos
por ayudarme a entender la vida con otros ojos.
Sergio Andrés Lagos Fuquen*

AGRADECIMIENTOS

Quiero agradecerle principalmente a mi mamá y a mi tío, que me han apoyado en todo este proceso de estudio, me han ayudado tanto económicamente como anímica mente a salir adelante durante todos estos años. A mis amigos y a mi novia por darme tantos momentos y recuerdos durante este tiempo compartido.

También quiero darle las gracias a mi director que gracias a su apoyo he aprendido grandes cosas y he podido llevar a cabo todo esto.

Le doy las gracias a la Universidad y a todos mis profesores que gracias a ellos me he llenado de intelecto y de cosas buenas tanto académicamente como personalmente.

Gracias a mis compañeros que han sido un apoyo gigante en todo este proceso, ya que de ellos también he aprendido grandes cosas. Muchas gracias por hacer esto realidad.

Mateo Parra Posse.

Agradezco a toda mi familia que me apoyaron y permitieron disfrutar esta parte de mi vida, solo quiero manifestar gratitud a mis padres por su guía, a mis abuelitos por su gran cariño, a mi hermano por animarme, a mi madrina y primo que son mi segunda familia y siempre me han dado amor, a mi novia por seguirme dando su apoyo y amor en este proceso.

Académicamente agradezco a los tutores por su tiempo, paciencia y dedicación. A mis amigos les agradezco por el ánimo y a mi compañero de tesis por apoyarnos mutuamente y siempre dándola toda para sacar este proyecto adelante.

Sergio Andrés Lagos Fuquen.

Índice general

Resumen	14
Introducción	15
1. Planteamiento del problema	16
2. Justificación	18
3. Impacto Social	20
4. Objetivos	21
4.1. Objetivo General	21
4.2. Objetivos Específicos	21
5. Antecedentes	22
6. Metodología	29
6.1. Investigación	30
6.2. Diseño	30
6.3. Desarrollo	31
6.4. Validación	31
7. Marco Teórico	32
7.1. Signos vitales	32
7.2. Temperatura	33
7.3. Frecuencia cardíaca	37
7.4. Oximetría	41
7.5. Microsoft .NET	44
7.6. Bases de Datos	44

7.7. MySQL	47
7.8. Plataforma Heroku	47
7.9. Plataforma Twilio	48
7.10. JavaScript	48
7.11. Raspberry Pi 3	49
8. Estado del Arte	50
8.1. An Integrated Healthcare Enterprise Information Portal and Healthcare Information System Framework	51
8.2. A review on open source architecture in Geographical Information Systems	51
8.3. Sistema de Alertas de Signos Vitales y de Ubicación a través de Dispositivos Móviles	52
8.4. Sistema web para el monitoreo de signos vitales y control del historial clínico de pacientes hospitalizados en servicios de medicina interna del hospital general regional "Dr. Rafael Pascasio Gamboa".	52
8.5. Tratamiento de rizados en la estimación de la densidad espectral de potencia de la señal de ritmo cardiaco.	53
9. Funcionamiento General del Sistema	54
10. Diseño y creación de Base de Datos	57
11. Diseño de Wireframes	59
12. Estructura del Back- End	68
12.1. Estructura del servidor	69
12.2. Conexión entre Heroku y MySQL	73
13. Estructura del Front-End	78
13.1. Parte visual de la aplicación web	79
13.1.1. Cliente con rol de Analista	80
13.1.2. Cliente con rol de Usuario	82
14. Generación de Alarmas	84
14.1. Plataforma Twilio	85
14.2. Mensaje de Alerta desde el código	86

15. Despliegue de la aplicación en Heroku	87
16. Procesamiento Base de Datos Physionet	89
17. Construcción Simulador	92
17.1. Creación de programa en C#	93
17.1.1. Envío de información a la página web:	93
17.1.2. Lectura de archivo txt, con la información	94
17.2. Integración solución con Hardware	95
17.2.1. Configuración Tarjeta Raspberry Pi 3	95
17.2.2. Paso de Computador desarrollador a Raspberry para ejecutar	96
18. Resultados	98
18.1. Visualizar	98
18.2. Almacenar	101
18.3. Generar alertas	102
Conclusiones	104

Índice de cuadros

7.1. Rangos de temperatura cuando se tiene hipotermia	36
7.2. Frecuencias cardíacas en condiciones normales dependiendo la edad	38
7.3. Zona de ritmo cardíaco en un 50 y 85 % de intensidad	39
7.4. Símbolos del Modelo E-R.	46
9.1. Rangos de Anomalías según signos vitales	55

Índice de figuras

5.1. Diagrama de partes que componen el proyecto mostrado en la conferencia.	24
5.2. Prototipo final y aplicación móvil.	25
5.3. Prototipo final y aplicación móvil.	26
5.4. Monitor de presión arterial KSBP2.	27
6.1. Diseño por etapas del proyecto.	30
7.1. Temperatura corporal en diferentes zonas	35
7.2. Señal continua (amarillar) y pulsátil (roja) de la absorción de los haces de luz	42
7.3. Absorción de la luz de sangre con oxihemoglobina y desoxihemoglobina	42
7.4. Curva de disociación de la hemoglobina	43
7.5. Modelo relacional.	47
7.6. Composición plataforma Heroku.	48
7.7. Raspberry Pi 3.	49
9.1. Diagrama general del funcionamiento del sistema de información.	56
10.1. Modelo Relacional	58
11.1. Mapa Web de la aplicación	59
11.2. Wireframe de la pestaña Home	60
11.3. Wireframe de la pestaña Login	61
11.4. Wireframe de la pestaña Signup	61
11.5. Wireframe del rol Analista	62
11.6. Wireframe del Historial de Datos	63

11.7.	Wireframe Datos en vivo	64
11.8.	Wireframe Historial de Alertas	64
11.9.	Wireframe del rol Usuario	65
11.10.	Wireframe del Historial de Datos	66
11.11.	Wireframe Datos en vivo	66
11.12.	Wireframe Historial de Alarmas	67
12.1.	Estructura del proyecto	72
12.2.	Ventana de Inicio de la plataforma Heroku	74
12.3.	Acceso a MySQL desde la terminal de Linux	77
13.1.	Pestaña inicial	79
13.2.	Pestaña del historial de datos del usuario	80
13.3.	Pestaña de visualización de datos en vivo del usuario	81
13.4.	Pestaña de historial de alertas del usuario	81
13.5.	Pestaña del historial de datos	82
13.6.	Pestaña de visualización de datos en vivo	82
13.7.	Pestaña historial de alertas	83
14.1.	Tabla de Anomalías tomada de la base de datos	84
14.2.	Mensaje de confirmación plataforma Twilio	85
14.3.	Credenciales de Twilio	86
16.1.	Ejemplo de formato de normalización de Signos Vitales	91
17.1.	Esquema general del simulador con el sistema.	92
17.2.	Interfaz Mobaxterm de la Raspberry	95
17.3.	Opción para publicar el proyecto desde visual studio	97
18.1.	Gráfica Frecuencia Cardíaca	99
18.2.	Gráfica Temperatura	99
18.3.	Gráfica Oxigenación en la Sangre	100
18.4.	Tabla de anomalías presentadas	100
18.5.	Tabla de Signos vitales de Usuarios	101
18.6.	Tabla de Usuarios	102
18.7.	Tabla de historial de alertas	102
18.8.	Mensaje Analista	103
18.9.	Mensaje Usuario	103

Índice de Códigos

12.1. Archivo .json del programa	69
17.1. Comandos de terminal para poder ejecutar .NET en la Raspberry Pi 3	96
18.1. Script creación Base de Datos.	113

Resumen

La presente investigación busca elaborar un sistema de información que permita almacenar y visualizar los signos vitales de diferentes usuarios como: frecuencia cardíaca, pulsioximetría (SpO2) y temperatura, los cuales se guardarán en una base de datos facilitando un historial de estas bio-señales que se podrán observar desde una página web desarrollada en Node.js. Además, este sistema cuenta con un módulo de alerta, la cual se acciona en el momento en que se perciba una anomalía en alguno de los signos nombrados anteriormente. Se crea un simulador de información apoyado en una base de datos de signos vitales públicas para la validación de este sistema.

El proceso inicia con un estado del arte, que permite la comprensión de por lo menos 3 signos vitales, sus valores típicos, sus irregularidades y qué resonancia tienen en el cuerpo. Se procede a construir el sistema de información partiendo del modelo de entidad relación que van a tener los datos almacenados, del desarrollo en la nube que se va a plantear para el manejo del sitio web del sistema, el diseño y desarrollo front end para cada usuario y el acople de todo este en una página pública. Una vez se tenga el sistema se hace la construcción de un simulador de información apoyado en una base de datos de signos vitales con el fin de comprobar la visualización y alertas de signos vitales enviados por el simulador.

Número de páginas: 112.

Introducción

La medicina a lo largo de su historia buscando salvar vidas, ha tenido avances significativos debido al mayor entendimiento del funcionamiento del cuerpo humano y su estructura, por ende, se ha permitido el aumento de la esperanza de vida en las personas a nivel mundial [1]. Ya que, a inicios del siglo XIX, ningún país tenía una expectativa de vida mayor de 40 años debido a factores tanto sociales como económicos [2].

En la actualidad, como consecuencia de la pandemia causada por el virus SARS-CoV-2 que generó la enfermedad llamada coronavirus (COVID-19), el monitoreo remoto ha permitido llevar un control y registro de diferentes casos respetando el aislamiento, generando una menor saturación en el sistema de salud [3].

De mano de la tecnología la ingeniería electrónica puede profundizar en el ámbito de la salud desarrollando una solución conjunta mediante campos como la bioingeniería, base de datos, microelectrónica y el Internet de las Cosas que influya en el desarrollo de una herramienta que permita al personal médico cumplir con el distanciamiento y permite mejorar el seguimiento con los pacientes; por otro lado el paciente tendría accesibilidad y portabilidad en la información de sus signos vitales por medio de una aplicación.

Capítulo 1

Planteamiento del problema

De acuerdo con el artículo “Cuál es el problema del sistema de salud colombiano” publicado en la revista Dinero en el año 2018, se menciona que: “Colombia es el país de América Latina con la cobertura más alta en salud, superando el 95 % de su población. Sin embargo, el 70 % de los afiliados está insatisfecho con el servicio según una encuesta del Banco Interamericano de Desarrollo (BID).” [4] Partiendo del gran porcentaje de insatisfacción por parte de los colombianos con el sistema de salud, se puede llegar a la conjetura que existen falencias en este sistema; cotidianamente se escucha que existen problemas económicos, administrativos, de organización, trámites y hasta de tiempo por consulta [5], sin embargo, el Ministerio de Salud y Protección Social manifiesta que: “el sistema de salud colombiano ofrece demasiado a sus usuarios, en particular da acceso a tratamientos relativamente recientes, a pesar que son tratamientos costosos y que, por otro lado, los colombianos pagan una fracción muy baja” [6]. Observando con esto, que Colombia puede mejorar este nivel de satisfacción implementando nuevas tecnologías en sus procesos, dado que unas de las causantes que se perciben como fallo en este sistema es la relación del personal médico y el paciente [7].

La tecnología ha ocasionado la transformación del mundo en estos últimos años y a Colombia puede llevarla a una época de cambios y soluciones en temas de salud. Para generar un cambio a través de la tecnología, se busca optimizar el seguimiento, almacenamiento y generación de alarmas de la captación de signos vitales, la cual sea escalable, que centralice la visualización de estos en un mismo sistema y que sea ligero en recursos de hardware para entidades pequeñas y medianas.

De lo anterior se genera la siguiente pregunta: ¿Qué herramienta electrónica se puede desarrollar para volver más eficiente el manejo, visualización y alertas de los datos médicos registrados de un paciente?

Para solucionar esta pregunta se plantea fabricar un sistema de información para visualizar y almacenar signos vitales de un ser humano, los cuales enviarán una alerta en el momento de presentarse anomalías; permitiendo portabilidad de historial, mejorando tiempos de respuesta ante anomalías de signos vitales y centralizando la visualización de varios signos en un solo sistema ya sea mostrando historial o captación en vivo.

Capítulo 2

Justificación

Un estudio hecho por la Oficina de Calidad del Ministerio de Salud y Protección Social muestra un ranking de importancia y satisfacción de los usuarios con los atributos de las Entidades Promotoras de Salud (EPS); esta encuesta se realizó a los diferentes usuarios de estas entidades en donde se quería sondear qué tan satisfechos estaban estos con los servicios administrativos, médicos, tiempos de espera en las salas, atención, agilidad en los diferentes servicios y personal médico entre otros. Como resultado de la encuesta, los usuarios concluyen que los temas de agilidad en los trámites, el buen trato del personal médico y la cercanía de los hospitales son cualidades favorables en el servicio de salud; en cuanto a la satisfacción personal, los usuarios encuentran como factor relevante el trato amable del personal asistencial, seguido por las instalaciones físicas de la EPS. Finalizado el análisis del estudio, la Oficina de Calidad muestra como resultado la importancia de indagar en los tiempos de espera para la atención, la simplicidad y agilidad en los trámites [8].

A partir de lo anterior, se evidencia la necesidad de implementar nuevas tecnologías prácticas para la simplicidad y agilidad en procesos internos entre paciente y centro de salud. Por este motivo se busca que a través de una herramienta en la nube se pueda crear una aplicación web, de fácil acceso, rápida y que permita al paciente o analista la captación y visualización de signos vitales para su almacenamiento y generación de alertas cuando se presenten anomalías.

Capítulo 3

Impacto Social

El alcance del proyecto impacta en el área de la salud, debido a que el sistema de información busca almacenar y visualizar los signos vitales además de generar alertas cuando se presenten anomalías. Usando la tecnología en pro de la mejora de procesos, el almacenamiento de signos vitales se hace en una base de datos alojada en la nube y la cual puede mostrar la información tomada a través de una página web de fácil acceso.

El área poblacional que se busca impactar directamente es a trabajadores del personal de salud y pacientes mayores de edad que necesitan un seguimiento de sus signos vitales constante. El fin del sistema de información es que los pacientes y los trabajadores de salud puedan automatizar la captación de datos clínicos y poderlos visualizar al tiempo, desde una plataforma web donde puedan estar chequeando sus signos vitales en tiempos específicos o en la actualidad y recibir alarmas de anomalías que se presenten.

Capítulo 4

Objetivos

4.1. Objetivo General

Diseñar un sistema de información que permita el almacenamiento y la visualización de signos vitales de diferentes usuarios, para la generación de alertas sobre anomalías de estos signos.

4.2. Objetivos Específicos

- Diseñar y desarrollar un sistema de información de datos que permita almacenar y visualizar signos vitales de usuarios con el fin de mantener un historial.
- Desarrollar un módulo de software en el sistema, para la generación de alertas en situaciones de anomalías en los signos vitales.
- Construir un simulador de información apoyado en una base de datos de signos vitales para validar el funcionamiento del sistema.

Capítulo 5

Antecedentes

Con el paso del tiempo, la sociedad descubre nuevas tecnologías o nuevos casos de enfermedades que los investigadores tienen que estudiar. El monitoreo de señales del cuerpo humano no es algo nuevo, es algo que a lo largo de la historia de la medicina se ha venido trabajando, descubriendo cada vez más progresos los cuales pueden ser de ayuda. En la actualidad, estos se ven complementados por sistemas de información hospitalaria (SIH o HIS según sus siglas en inglés) los cuales buscan la optimización en el manejo y tratamiento de datos tanto administrativos como de salud para mejorar la gestión hospitalaria.

Los avances tecnológicos han llevado a que estos monitores ya no sólo se encuentren en hospitales o centros de salud ya que, ahora existen los smartwatch o relojes inteligentes los cuales, en su mayoría, traen contadores de pasos, miden el ritmo cardíaco, la presión arterial, entre otros. Además, ya se encuentran en el mercado monitores digitales, los cuales el paciente puede usar desde su casa o cualquier parte en donde se encuentre, generando que estos lleven un control particular de su salud, pero desconociendo la opinión de expertos.

Además de proyectos académicos, los sistemas de salud han pasado a ser aplicaciones comerciales personalizables, cómo lo son las plataformas donde se registra información de la salud del usuario; un ejemplo de esto es: "Google Fit"[9] donde el almacenamiento y la visualización de signos vitales e información de cuidado de la salud, pueden estar al alcance del usuario final con la única condición de que se tenga algún dispositivo que sea compatible con la aplicación y esté en constante conexión para hacerle seguimiento de los datos captados.

Los siguientes son proyectos enfocados a los sistemas de información, personal y hospitalarios; dado que es el sitio donde se quiere implementar en el sistema:

Sanabria Fredy y Romero Rafael (2005), Universidad de los Andes de Bogotá D.C.

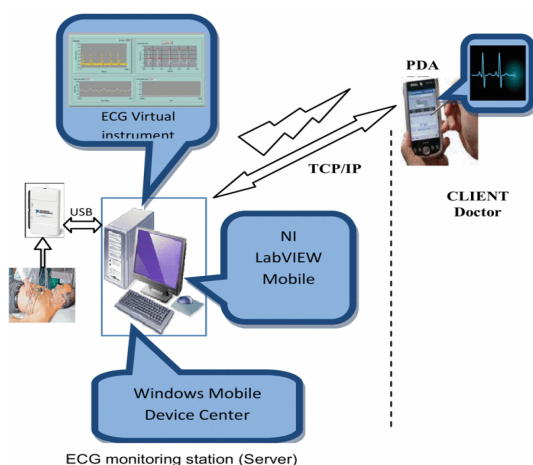
En su tesis titulada: "Diseño e implementación de un sistema para el almacenamiento de historial clínico en tarjetas inteligentes". Instauraron un sistema compuesto de un dispositivo lector - escritor de tarjetas inteligentes con doble seguridad (cifrado y firma electrónica), un portal de internet para la visualización de datos que permite almacenar y consultar los historiales clínicos, y un lector portátil para emergencias [10].

De este proyecto se puede observar un método de visualización y almacenamiento de datos útil y funcional para el registro de información médica en un sitio web, el cual posee una base de datos para almacenar los informes generados. Los inconvenientes presentados en este proyecto se basan más que todo en el hardware ya que el cronograma propuesto varió en gran medida por temas de adquisición y disponibilidad de tarjetas de desarrollo, lo que indica que este es un factor importante a la hora de diseñar y desarrollar un proyecto.

Conferencia E-Health y Bioingeniería (2011), Iasi, Romania

En esta conferencia se presentó un proyecto de mayor robustez, en donde se presenta un sistema con una inteligencia artificial que monitorea señales biológicas y transmite información relevante de forma remota a una aplicación portátil la cual está basada en la arquitectura de cliente-servidor; esta monitorea las bioseñales para establecer un diagnóstico previo y dependiendo de esto ayudar a los pacientes críticos a acelerar un diagnóstico previo (ver figura 5.1) [11].

Figura 5.1. Diagrama de partes que componen el proyecto mostrado en la conferencia.



Fuente:

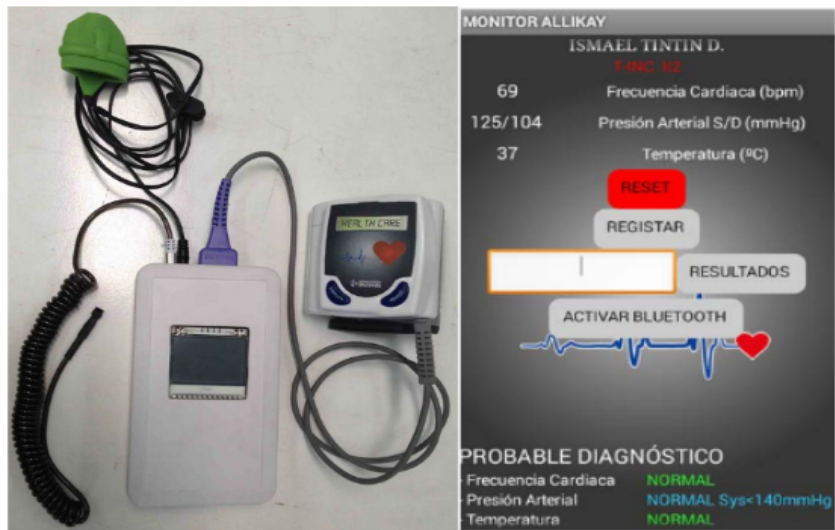
<https://www.researchgate.net/publication/254051913-Wireless-intelligent-systems-for-biosignals-monitoring>

Aunque el sistema está enfocado para asistente personal (PDA), al ser necesario un computador host local para poder ingresar al sistema de monitoreo y análisis, este va a generar gastos de cuidado de hardware y puede presentarse problemas de portabilidad de información generando incidentes a futuro en el manejo de los datos.

Tintín Edison (2015), Universidad Politécnica Salesiana de Cuenca, Ecuador.

En el trabajo: “Diseño y elaboración de un Prototipo de monitor de signos vitales aplicando métodos no invasivos con comunicación de datos a dispositivos móviles” [12] se elabora un prototipo (ver figura 5.2) en el cual obtiene presión arterial (brazalete comercial), frecuencia cardíaca (tarjeta electrónica que adquiere y procesa la señal) y temperatura corporal (LM35), los recopila en una tarjeta electrónica ChipKIT la cual los procesa y se envían por medio de bluetooth a una aplicación móvil para dispositivos Android y generar un diagnóstico o historial clínico. También los signos vitales obtenidos son mostrados en pantalla LCD y el precio tentativo para la venta era de \$442,04 USD.

Figura 5.2. Prototipo final y aplicación móvil.



Fuente:

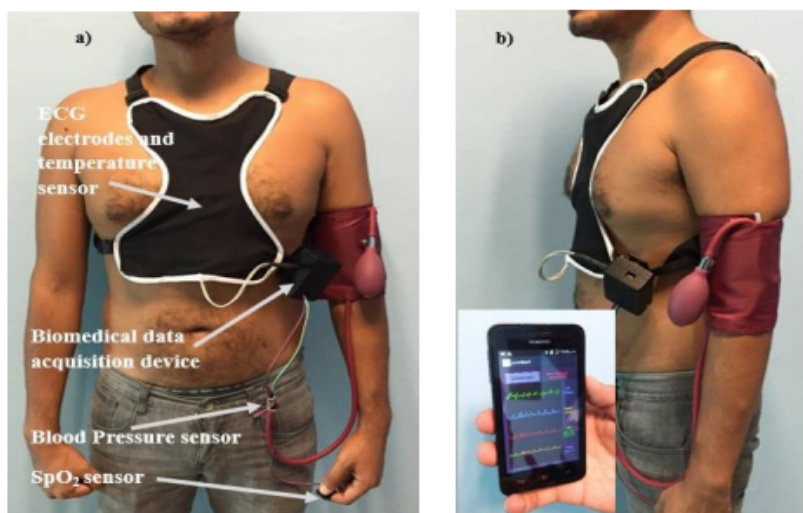
<http://bibliotecadigital.usb.edu.co/bitstream/10819/1411/1/Monitor-Signos-Vitales-Parra-2011.pdf>

Ya que la aplicación móvil sólo está disponible para dispositivos Android, eliminando la opción de visualizar los datos en diferentes plataformas.

Rebolledo Zaic (2016), Universidad Autónoma de Guerrero de Chilpancingo de los Bravo, México.

En el trabajo: “Monitor de signos vitales portátil” [13] se elabora un prototipo portátil (ver figura 5.3) capaz de adquirir una señal electrocardiográfica, pulso cardíaco, presión arterial, SpO₂, temperatura y niveles de glucosa en la sangre de manera no invasiva a lo largo de ocho (8) horas, logrando procesarlas a través de un Arduino y enviarlas por medio de bluetooth a una aplicación móvil de Android para ser mostrados gráficamente. El éxito del sistema esta en la estructura modular , es decir, que si uno de los sensores no se desea poner, este no repercute en la información que toma el sistema, únicamente no se muestra; se tuvo muy en cuenta la economía del sistema, dado que este no emplea una base de datos, sino que está enfocado a un registro de información en un archivo plano; generando que la expansión del producto se vea afectado o que deba modificarse en el momento de integrar una base de datos para organizar la información.

Figura 5.3. Prototipo final y aplicación móvil.



Fuente:

<http://ri.uagro.mx/handle/uagro/295>

Óscar Sarmiento y John Edilberto Rubio (2018), Universidad Distrital.

En el trabajo “Monitoreo remoto de signos corporales y transmisión de datos y alertas a una aplicación instalada en un smartphone” [14] es un prototipo para llevar a cabo propuestas orientadas a patologías de las personas con la cual mide diferentes signos vitales, y posee una aplicación móvil en la cual administra y muestra los datos recibidos vía bluetooth, además de generar alertas dentro de esta misma cuando los signos vitales sobrepasan los umbrales normales.

Monitor de presión arterial KSBP2 (2018).

Es un brazalete fabricado por la empresa Koogeek (ver figura 5.4) el cual mide la presión arterial y muestra los resultados en una aplicación gratuita para iOS y Android, con la cual se comunica por medio de WiFi y bluetooth, permite administrar datos de usuarios múltiples en la nube (máximo 16) y posee una pantalla LCD que muestra continuamente los signos vitales. [15]. El seguimiento se vuelve autónomo del usuario y no hay una unión de frente con algún ente de salud, generando que no haya suficiente trazabilidad a nivel de sistema salud.

Figura 5.4. Monitor de presión arterial KSBP2.



Fuente:

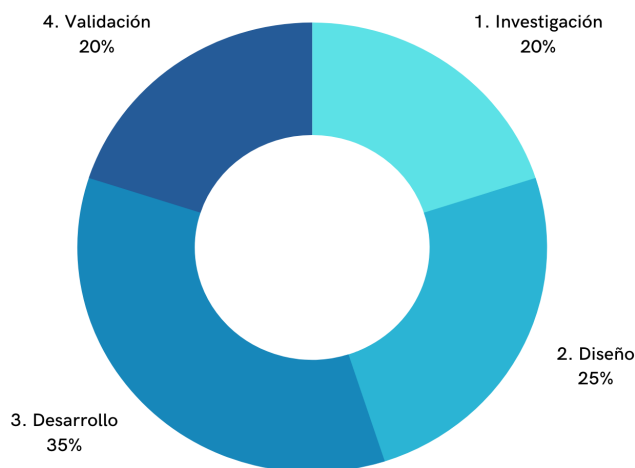
<https://www.koogeek.com/smart-health-3076/p-ksbp2.html>

El presente proyecto busca plantear mejoras partiendo de los proyectos mencionados anteriormente, con el fin de generar un sistema de información versátil, que no dependa de un solo Sistema Operativo para ejecutarse, sin gasto económico ni limitantes operacionales con hardware, que permita adaptarse a un entorno de salud donde se incluya al personal médico como analistas de información de los datos suministrados por los pacientes e innovar con una alerta distinta a las observadas.

Capítulo 6

Metodología

Como se puede observar en la figura 6.1 la ejecución del presente proyecto se realiza a partir de la metodología experimental la cual se inicia con una investigación de los signos vitales, sus formas de obtención y las anomalías de estas en posibles situaciones de alerta. Se plantea el funcionamiento general del sistema de información, posterior a esto se diseña el modelo de Entidad - Relación de la base de datos y se definen unos wireframes que definen el diseño de la pagina web. Teniendo claro la base del proyecto y lo que se busca tener, se desarrolla tanto back-end como front-end del sistema de información, para la conexión con el sistema de almacenamiento en la nube a través de Heroku y la configuración de alarmas en el sistema. Para la validación, se crea un simulador de información de signos vitales basado en una base de datos pública (Physionet) para analizar la funcionalidad del sistema.

Figura 6.1. Diseño por etapas del proyecto.

Fuente: Propia

6.1. Investigación

En esta primera etapa, se realiza una búsqueda de ¿Qué información aportan estas bioseñales para la salud de una persona?, ¿Cuáles son las anomalías que se pueden presentar en estos signos vitales? y ¿En cuáles partes del cuerpo se pueden tomar mejor estas señales?. Para solventar estas dudas, se realiza un estudio en bases de datos de investigación médica y revistas relacionadas con la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), con el fin de tener información verídica.

6.2. Diseño

En esta etapa se establece el funcionamiento general del sistema de información para determinar estructuras internas y diseños generales contemplados en el modelo Entidad-Relación de la base de datos y en los

wireframes que representan el diseño estándar de la página web. El modelo de entidad relación se realiza a través de la plataforma Lucidchart con el fin de posteriormente se genere en un gestor de base de datos. Para el diseño de los wireframes se utiliza la herramienta online InVision para la ilustración de las diferentes rutas de la aplicación web.

6.3. Desarrollo

En esta tercera etapa, mediante el uso del entorno de ejecución Node.js basado en el lenguaje de programación JavaScript (JS) se despliega la estructura del servidor usando el framework Express.js. También se desarrolla el entorno gráfico a través del framework Bootstrap que incorpora elementos CSS y JS para manejar el diseño y la presentación del HTML de la aplicación web. Por último, se realiza el despliegue de todo el sistema a la nube, a través de la plataforma Heroku.

6.4. Validación

Por último se crea un simulador de información de signos vitales fundamentado en una base de datos pública (Physionet) la cual va a estar conectada al sistema generado con el fin de suministrar datos para visualizar, almacenar y generar alertas si existe alguna anomalía. Este simulador, se va a ejecutar en una Raspberry pi 3 modelo B, la cual ejecutará una aplicación de consola .Net con lenguaje C para enviar datos al sistema de información ya desplegado en la nube.

Capítulo 7

Marco Teórico

7.1. Signos vitales

Son señales bioeléctricas, las cuales reflejan funciones esenciales del cuerpo, como lo son el ritmo cardíaco, la presión arterial, la frecuencia respiratoria y la temperatura corporal. Estos signos vitales cambian dependiendo varios factores como lo son: la edad, el peso, el sexo, la cantidad de ejercicio que haga la persona, entre otras. Los rangos normales para un adulto sano promedio son:

- Presión arterial: 90/60 mm Hg hasta 120/80 mm Hg.
- Respiración: 12 a 18 respiraciones por minuto.
- Pulso: 60 a 100 latidos por minuto.
- Temperatura: 97.8°F a 99.1°F (36.5°C a 37.3°C)
promedio de 98.6°F (37°C).

Estos rangos de medida pueden variar por diferentes factores o enfermedades que presente la persona [16]. El monitoreo de signos vitales es muy importante para detectar problemas de salud ya sea en un centro hospitalario o desde la residencia de cada persona [17].

7.2. Temperatura

La temperatura es fundamental para proporcionar vida, este parámetro puede afectar considerablemente en los metabolismos, debido a que las altas o bajas temperaturas pueden traer consecuencias o enfermedades en el cuerpo humano, ya sean casos de hipotermia (bajas temperaturas) o hipertermia (altas temperaturas). Estas alteraciones son principalmente detectadas en adultos mayores dado que estos son especialmente susceptibles por ser polimedicados dado que todo medicamento genera efectos secundarios y factores propios como alteraciones cognitivas, motrices o infecciones desarrolladas que repercuten en alteraciones hormonales [18].

Para que el cuerpo pueda responder a la alteración de la temperatura corporal, el termorregulador biológico con el cual contamos se encuentra en el hipotálamo y tiene entre sus funciones principales regular además de la temperatura corporal, la presión sanguínea, la tensión muscular, los niveles de hambre, entre otros. Los cambios de temperatura provocan la respuesta neuronal de los receptores cutáneos y la variación en la temperatura sanguínea lo cual le facilita al hipotálamo generar una respuesta adecuada y mantener la temperatura corporal en sus límites normales [19]. En los casos de temperatura alta, se comprobó que el baño de agua tibia a 37°C es la mejor manera de ayudar de manera externa al cuerpo para termoregularse [20].

Para poder captar la temperatura a lo largo del tiempo diferentes instrumentos y métodos que se han venido desarrollando para captar este signo vital de manera invasiva o no y en diferentes ubicaciones

algunos ejemplos son: [21]

- Medición de la temperatura cutánea axilar
- Medición de la temperatura cutánea de la frente
- Termómetro en la arteria temporal
- Medición oral
- Mediciones de la temperatura corporal central a través de la membrana timpánica
- Medición rectal
- Nasofaringe
- Esófago
- Tecnología de doble sensor / flujo de calor

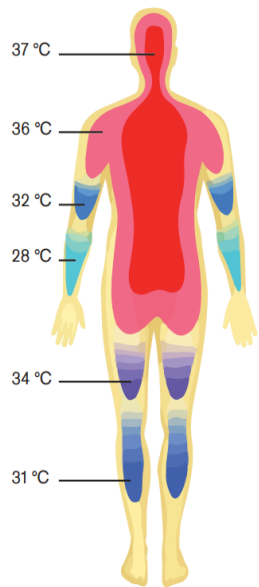
La temperatura corporal varía dependiendo de la zona en la que sea medida (Ver figura 7.1), por ejemplo, en las zonas abdominales, en el sistema torácico y sistema nervioso la temperatura es de 2° a 4 ° centígrados más que en las extremidades del cuerpo [22].

La temperatura en las zonas abdominales, torso y cabeza es de 37° centígrados aproximadamente, en cambio el rango de temperatura de las extremidades se encuentra entre 27° y 32° centígrados para una persona adulta sin ropa en condiciones normales [21], estos rangos de temperatura se dan en condiciones normales de la persona, ya que debido a algunas enfermedades o infecciones en el cuerpo la temperatura puede subir o bajar generando cambios en los diferentes sistemas del cuerpo humano. Dos consecuencias comunes que afectan directamente la temperatura corporal son la hipotermia y la hipertermia.

- **Hipotermia**

Hace referencia al enfriamiento de las zonas del cuerpo por debajo

Figura 7.1. Temperatura corporal en diferentes zonas



Fuente: <https://www.draeger.com/Library/Content/t-core-bk-9101301-es-1604-1.pdf>

del rango normal (Los rangos de hipotermia se pueden observar en el cuadro 7.1), en donde el cuerpo comienza a reaccionar con algunos temblores en diferentes zonas del cuerpo, además comienzan a aparecer diferentes síntomas como lo son el aumento del ritmo cardiaco, mayor frecuencia respiratoria, problemas de coordinación y una reducción del volumen sanguíneo [21]. La hipotermia puede ser causada por algún tipo de accidente en donde la persona se ve expuesta a bajas temperaturas por un tiempo prolongado, o por alguna intervención médica en donde todo está controlado.

Cuadro 7.1. Rangos de temperatura cuando se tiene hipotermia

Temperatura	Clasificación
33 - 36 °C	Hipotermia leve
28 - 32 °C	Hipotermia, reducción del metabolismo, depresión respiratoria, pérdida de conocimiento
Menor a 28 °C	Hipotermia grave, fallo del sistema termorregulador, fibrilación ventricular, rigidez, falta de reacción en las pupilas, bradicardia, paro cardíaco

Fuente: <https://www.saludsavia.com/contenidos-salud/enfermedades/hipotermia>

La hipotermia deja consecuencias en el cuerpo ya que afecta todo el organismo como lo son el estrés y sensación de frío constante, se puede generar resistencia a algunos medicamentos como lo son la insulina, infecciones de heridas, cicatrización deficiente de heridas, además de otras múltiples consecuencias.

■ **Hipertermia**

También llamada comúnmente fiebre, es un estado en el cual la temperatura corporal incrementa por encima de los límites normales. Se considera hipertermia a temperaturas superiores de 37.5 °C. La hipertermia se puede causar por razones infecciosas o no infecciosas.

- *Causas de fiebre no infecciosas:*
 - Abstinencia de alcohol o drogas.
 - Fiebre después de una transfusión.
 - Insuficiencia suprarrenal.
 - Infarto de miocardio.
 - Pancreatitis.
 - Trombosis venosa profunda.
 - Embolia pulmonar.
 - Hematomas.
 - Cirrosis.
 - Úlceras de decúbito.

- *Causas de fiebre infecciosas:*
 - Neumonía y sinusitis.
 - Sepsis relacionada con cateterismo.
 - Diarrea por *C. difficile*.
 - Sepsis abdominal e infecciones de heridas.

Cabe recalcar que este aumento de temperatura es un cambio fisiológico que realiza el cuerpo para defenderse de infecciones y enfermedades que ingresan al organismo. Aunque, es un mecanismo de defensa natural, esto puede generar daños internamente; por este motivo se han venido presenciando modelos de predicción de picos de fiebre para poder anticipar a través de monitoreo constante y modelos matemáticos, desgastes en el organismo; para la Unidad de Cuidados Intensivos [23].

7.3. Frecuencia cardíaca

En términos simples este signo vital puede definirse como la cantidad de veces que un corazón late (realiza contracción y relajación) por minuto. Algunas investigaciones han determinado que con esto se puede validar la variación en el tiempo entre los intervalos R y R medidos en un electrocardiograma (ECG) los cuales pueden ayudar a detectar problemas coronarios y enfermedades cardiovasculares [24]. La frecuencia cardíaca puede variar dependiendo factores como lo son el ejercicio, diferentes emociones o alertas a la seguridad de la persona. Los institutos nacionales de salud de los Estados Unidos han publicado una lista de frecuencias cardíacas normales en reposo dependiendo la edad de la persona (Ver tabla 7.2)[25].

Cuadro 7.2. Frecuencias cardíacas en condiciones normales dependiendo la edad

Edad	Frecuencia Cardíaca normal (ppm)
Hasta 1 mes	70 - 190
De 1 a 11 meses	80 - 160
Entre 1 y 2 años	80 - 130
Entre 3 y 4 años	80 - 120
Entre 5 y 6 años	75 - 115
Entre 7 y 9 años	75 - 110
Más de 10 años	60 - 100

Fuente: <https://www.medicalnewstoday.com/articles/es/291182#frecuencias-cardiacas-ideales-durante-el-ejercicio>

Una persona que realiza ejercicio frecuentemente debe tener en cuenta que la frecuencia cardíaca cambia dependiendo de la actividad física que se realice, dado que estas actividades piden una mayor oxigenación en el cuerpo para proporcionar más energía.

La Asociación Americana del Corazón establece que la frecuencia cardíaca máxima durante el ejercicio debe ser de aproximadamente 220 ppm menos la edad de la persona [25]. Debido que cada organismo reacciona diferente frente a la actividad física la frecuencia cardíaca no debe ser un número exacto como tal, sino que debe estar entre un rango de actividad dependiendo la intensidad del ejercicio.

Cuadro 7.3. Zona de ritmo cardíaco en un 50 y 85 % de intensidad

Edad	Zona de ritmo cardíaco 50 a 85 por ciento de esfuerzo (ppm)
20	De 100 a 170
40	De 90 a 153
60	De 80 a 136
70	De 75 a 128

Fuente: <https://www.medicalnewstoday.com/articles/es/291182frecuencias-cardiacas-ideales-durante-el-ejercicio>

Para poder obtener la frecuencia cardiaca, ya sea en reposo o cuando se ha hecho alguna actividad física, existen varios métodos de medida, unos convencionales y otros más sofisticados. La forma más tradicional de medición es contar cuantas veces late el corazón en un minuto, para esta medida es necesario encontrar una parte del cuerpo donde se pueda sentir el pulso cardiaco y se mide con los dedos de la mano exceptuando el dedo pulgar ya que en éste se puede sentir el pulso cardiaco y se generan confusiones en la medida; las partes del cuerpo más comunes para encontrar y sentir más fácilmente los latidos del corazón son [26]:

- **Carótida:** se explora la zona del cuello pasa la arteria carótida, es la más común para tomar el pulso cardiaco.
- **Radial:** se explora la arteria radial, y se palpa en la zona de la muñeca, en la cara anterior y lateral a esta.
- **Humeral o Braquial:** hay que palpar la parte interna del brazo exactamente en el pliegue del codo, que es donde pasa la arteria braquial.
- **Temporal:** se palpa la parte de la sien en la zona temporal ubicada delante de la oreja por donde pasa la arteria temporal.
- **Tibial posterior:** en esta zona se encuentra la arteria tibial posterior, y se tiene que palpar por detrás del maléolo interno y el tendón de Aquiles.

Una forma mucho más sofisticada y tradicional es por medio de un monitoreo electrocardiográfico ambulatorio de 24 horas, también llamado Holter. Por medio de éste se obtienen señales de la frecuencia cardíaca digitalizadas para que posteriormente sean analizadas por un experto de la salud; por esta razón actualmente se han venido usando tecnologías mucho más legibles y fáciles de entender, como lo es el pulsómetro digital. Existen varios métodos para medir la variación de la frecuencia cardíaca, los cuales en un inicio fueron en el dominio del tiempo, pero al ver que estos no especificaban muy bien los datos, se estudiaron nuevos métodos, pero esta vez en el dominio de la frecuencia. Los latidos del corazón generan una señal periódica, es decir, que no varían en el tiempo, y es por esta razón que es mucho más fácil encontrar variaciones en la señal en el dominio de la frecuencia [27].

La variación de la frecuencia cardíaca es alterable por edad, respiración, fluctuación de la presión arterial y temperatura [28], esto permite darle mayor relevancia a la información de este signo vital el cual termina relacionando varios aspectos internos del cuerpo y brindar más información a nivel general del cuerpo. Tal ha sido su importancia que desde inicio de la pandemia se han venido desarrollando soluciones que integren la ECG para poder captar este signo, como por ejemplo se ha podido desarrollar hasta en un aplicativo móvil [29].

Una de las anomalías comunes que puede presentar este signo vital son las arritmias cardíacas, las cuales son latidos irregulares del corazón y ocurren cuando los impulsos eléctricos que coordinan los latidos del corazón no funcionan adecuadamente [30]. Las arritmias se pueden clasificar en dos:

- Taquicardia: es cuando la frecuencia cardíaca en reposo de una persona es superior a los 100 latidos por minuto. Los tipos de taquicardia son los siguientes:
 - Fibrilación auricular
 - Aleteo auricular

- Taquicardia supraventricular
 - Fibrilación ventricular
 - Taquicardia ventricular
- Bradicardia: se presenta cuando la frecuencia cardíaca en reposo es inferior a los 60 latidos por minuto. Los tipos de bradicardia son:
- Síndrome del seno enfermo
 - Bloqueo de la conducción

7.4. Oximetría

La oximetría se basa en los principios fisiológicos de que la hemoglobina oxigenada y desoxigenada los cuales tienen diferente espectro de absorción [31].

La hemoglobina (HB) según Peñuela: “son las proteínas globulares, que fijan oxígeno en los pulmones y lo transportan por la sangre hacia los tejidos y células que rodean el lecho capilar del sistema vascular. Al volver a los pulmones, desde la red de capilares, la hemoglobina actúa como transportador de CO₂ y de protones” [32].

La definición anterior da a entender que la oximetría, permite conocer el adecuado funcionamiento respiratorio y circulatorio, esto se logra mediante la fotoplestimografía, la cual según De la Peña Sanabria: “Es una técnica óptica no invasiva que determina el seguimiento de las pulsaciones asociadas con los cambios en el volumen sanguíneo.” [33], esto se realiza mediante dos haces de luz distintos (roja e infrarroja) que atraviesan la extremidad medida (dedo índice o lóbulo de la oreja ya que debe ser una parte del cuerpo traslucida y con buen flujo sanguíneo) y que luego son recibidas por un fotodetector que analiza la absorción de cada una.

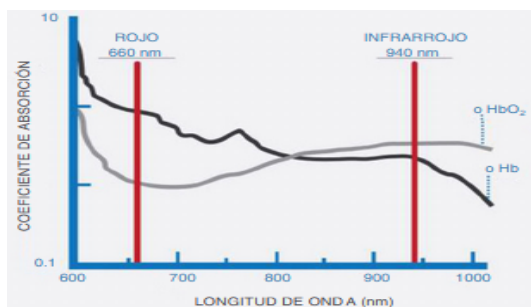
Posteriormente separa entre continua y la pulsátil (sangre arterial que presenta absorbancia variable en el tiempo) (Ver figura 7.2) denominada antes como onda fotopletismográfica la cual posee información de volumen sanguíneo en el movimiento de sístole y diástole, índice de saturación de oxígeno, frecuencia cardíaca y curva del pulso.

Figura 7.2. Señal continua (amarillar) y pulsátil (roja) de la absorción de los haces de luz



Fuente: Propia

Figura 7.3. Absorción de la luz de sangre con oxihemoglobina y desoxihemoglobina



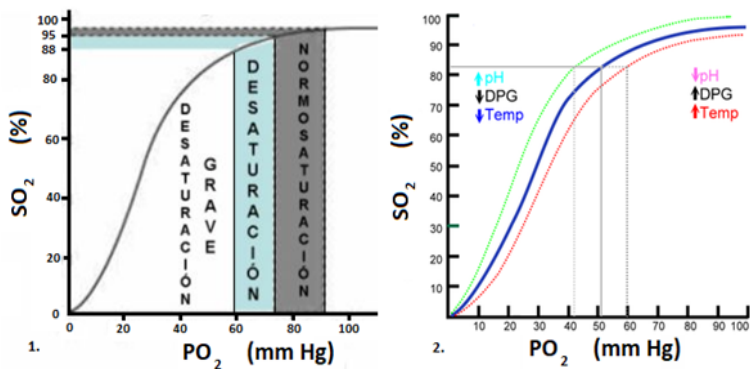
Fuente:

<https://www.elsevier.es/es-revista-perinatologia-reproduccion-humana-144-articulo-indice-perfusion-periferica-uci-neonatal-S0187533717300584#imagen-2>

Se usa la luz roja y luz infrarroja debido a que cuando la sangre posee hemoglobina oxigenada (HbO₂) presenta una mayor absorción de la luz infrarroja y cuando tiene hemoglobina desoxigenada (Hb) tiene mayor absorción de luz roja (Ver figura 7.3). De acuerdo con el radio de la absorbancia de cada una, se correlaciona con la proporción de hemoglobina saturada y desaturada en circulación y saca la saturación de la hemoglobina con la Ecuación 7.1 (Porcentaje de saturación de oxígeno), la cual por medio de la curva de disociación de la hemoglobina se puede relacionar la saturación con la presión del oxígeno y se puede determinar rangos de normosaturación (mayor de 95 %), desaturación leve (SO₂ entre 93 % y 95 %), desaturación moderada (SO₂ entre 88 % y 92 %) y desaturación grave (menor al 88 %) (ver figura 7.4.)

$$SaturaciónO_2 = \frac{HbO_2}{(Hb + HbO_2)} \tag{7.1}$$

Figura 7.4. Curva de disociación de la hemoglobina



Fuente (a): <https://image-api.onlineeducation.center/v2/image/max-width/800/statics/articles/public/524-02.jpg>

Fuente (b): <https://es.wikipedia.org/wiki/Efecto-Bohr#/media/Archivo:Oxyhaemoglobin-dissociation-curve.png>

7.5. Microsoft .NET

Es un Framework de ejecución administrado para Windows que consta de Common Language Runtime (CLR), que es el motor de ejecución que controla las aplicaciones en ejecución, y la biblioteca de clases, que proporciona una biblioteca de código probado y reutilizable al que pueden llamar desde sus propias aplicaciones [34]. Es usado principalmente para la creación y ejecución de servicios web y aplicaciones de internet.

7.6. Bases de Datos

Las bases de datos en la actualidad es un tema de suma importancia, ya que gracias a ellas es posible administrar de una forma organizada los datos que requieran ser guardados dependiendo del caso en el que sea necesario. Las bases de datos son un conjunto de datos interrelacionados que son almacenados en un ordenador, los cuales pueden variar con el tiempo. Una base de datos debe ser capaz de obtener la información de una forma sencilla, rápida y confiable, además estos datos deben tener coherencia, no contradecirse entre ellos y sobre todo se deben tener certeza de la seguridad y almacenamiento de estos mismos [35]. Algunos ejemplos de aplicación en diferentes sectores de la economía son los siguientes:

- Sectores Bancarios: para almacenar y guardar la información privada de los clientes de sus cuentas bancarias, préstamos, transacciones y demás
- Universidades: guardar información de los estudiantes como lo son las matrículas, semestre actual, materias ya cursadas y por cursar, entre otras.

- Empresas de Telefonía: para poder tener un registro claro de las llamadas de los usuarios, la generación mensual de su factura, almacenar el registro de su plan de datos (en dado que se tenga).
- Aerolíneas: poder guardar información de los clientes y su compra de tiquetes aéreos, además de eso, tener un registro de los vuelos nacionales e internacionales diariamente.

Estas son algunas aplicaciones para las que se pueden llegar a necesitar una base de datos con el fin de tener un registro, almacenamiento y seguridad en sus datos [36]. Así como las bases de datos traen ventajas como independencia de datos, mayor valor informativo, compartición de los datos, mejora de accesibilidad e integridad de los datos; también traen algunas desventajas como lo son el costo de instalación de una base de datos, el personal especializado en el tema, la baja estandarización, entre otros [37].

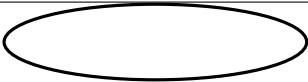
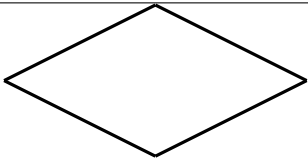
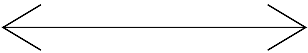
Para mantener una organización en las bases de datos es muy importante cumplir con una serie de requisitos que pueden denominarse los principales inconvenientes como lo son: la redundancia de datos, problemas de integridad y atomicidad, anomalías en el acceso concurrente, entre otros; estos problemas se pueden presentar frecuentemente a la hora de diseñar una base de datos y hay que tenerlos en cuenta. [38]. Para evitar los principales inconvenientes anteriormente mencionados lo mejor es realizar un modelo en el cual se puedan describir los datos mediante ilustraciones, relaciones, figuras y demás. Los modelos de datos más conocidos son los siguientes [39]:

- **Modelo Entidad-Relación**

Está basado en una percepción del mundo real que consta de objetos (entidades) y las conexiones (relaciones) que existen entre ellos. Los objetos poseen ciertos atributos, propios de dicho objeto, y las relaciones son las encargadas de conectar unos objetos con otros que tengan dependencia mutua o que sólo uno dependa del otro. Este modelo cuenta con una serie de objetos los cuales ayudan

a describir gráficamente la base de datos como se muestra en el cuadro 7.4.

Cuadro 7.4. Símbolos del Modelo E-R.

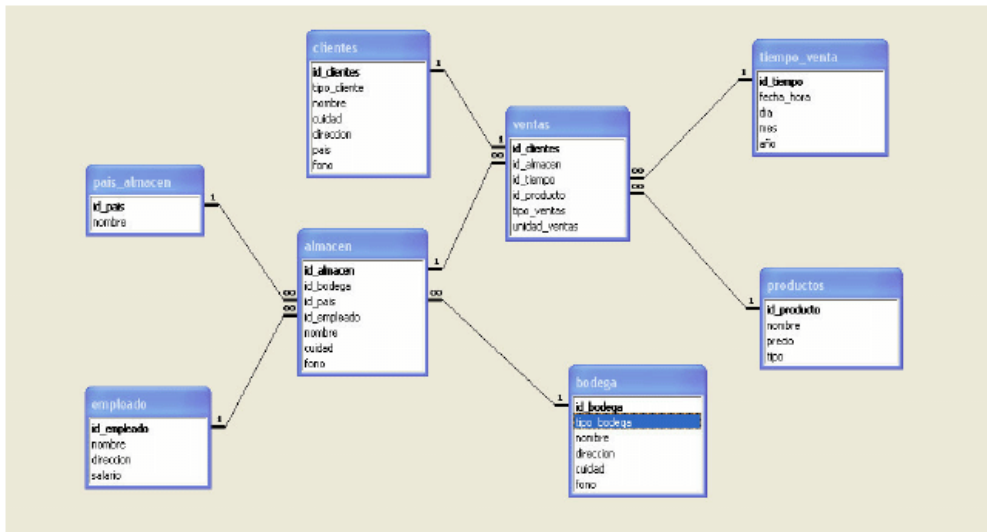
Símbolo	Acción
	Representa el conjunto de entidades
	Representa los atributos que tienen las entidades
	Representa la relación entre los diferentes conjuntos de entidades que hay en el modelo
	Representan la unión de los atributos con el conjunto de entidades y viceversa

Fuente: Propia

■ **Modelo Relacional**

Este modelo no es tan gráfico como el modelo E-R, éste usa tablas para representar los datos y las relaciones entre ellos (ver figura 7.5). Cada tabla que se maneja representa un objeto que va a tener la base de datos, y cada tabla va a contener un conjunto de propiedades o atributos los cuales van a ser únicos y se recomienda que ninguno tenga el nombre de la clase u objeto. Es un modelo basados en registros y es el más usado a nivel mundial teniendo una gran cantidad de bases de datos basados en este modelo.

Figura 7.5. Modelo relacional.



Fuente:

https://www.researchgate.net/figure/Modelo-relacional-del-sistema-Operacional-BD-fuente_fig1_221419392

7.7. MySQL

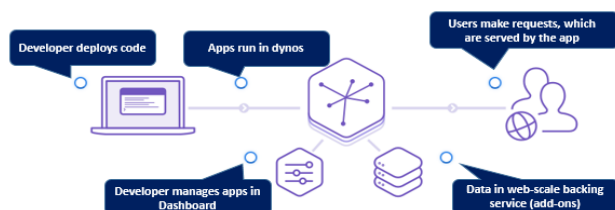
Es un sistema de datos relacional, desarrollado por Oracle Corporation y es considerada una de las bases de datos de código abierto más importantes a nivel mundial actualmente, y una de las más populares junto con Oracle y Microsoft SQL Server en el desarrollo web.

7.8. Plataforma Heroku

Es una plataforma que presta servicios en la Nube la cual soporta distintos lenguajes de programación como Java, Node.js, Python, PHP, Scala, entre otros. La base de datos de su sistema operativo es basada en Debian o en sus más recientes versiones es basada en Debian Ubuntu [40]. Las aplicaciones son ejecutadas en contenedores Linux ligeros que

se llaman "dynos" (ver figura 7.6), esta plataforma ofrece diferentes tipos de dynos para la finalidad que se requiera [41].

Figura 7.6. Composición plataforma Heroku.



Fuente:

<https://ifgeekthen.nttdata.com/es/Que%20es%20Heroku%20y%20principales%20características>

7.9. Plataforma Twilio

Es una plataforma de desarrollo que permite a los programadores construir aplicaciones mediante API's alojadas en la nube las cuales se integran en sistemas web de la plataforma para comunicarse a nivel mundial. Twilio cuenta con servicios como: Mensajería, Llamadas de Voz y Videollamadas [42].

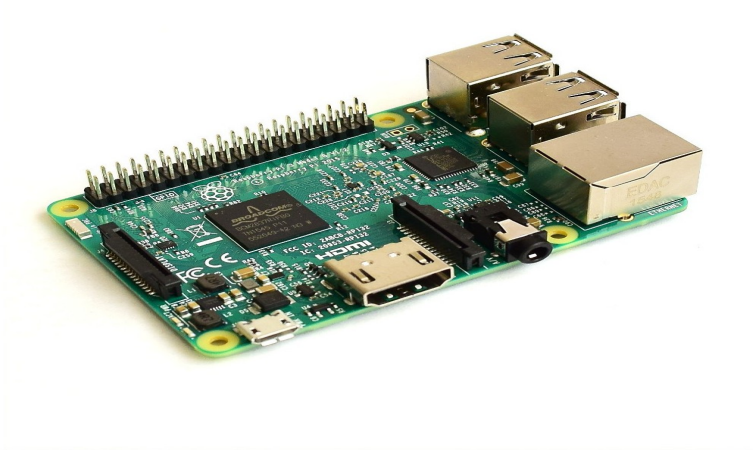
7.10. JavaScript

Es uno de los lenguajes de programación más usados actualmente para las aplicaciones web, permite controlar el comportamiento de las páginas y la interacción con los clientes que la utilizan, además de permitir la comunicación asíncrona dentro de sus desarrollos. JavaScript no sólo es exclusivamente para trabajar en desarrollos web, este lenguaje cuenta con varios frameworks para el desarrollo de servidores como lo son: NodeJS, ExpressJS, React entre otros [43].

7.11. Raspberry Pi 3

Es una tarjeta (ver figura 7.7) que cumple la función de un ordenador con LAN inalámbrica y conectividad Bluetooth; con Broadcom BCM2837, Cortex-A53 (ARMv8) 64-bit SoC y una GPU Videocore IV [44]. Las cuales permiten la recreación de un computador de 64 bits a un bajo costo de aproximadamente \$170.170 COP [45]

Figura 7.7. Raspberry Pi 3.



Fuente:

<https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>

Capítulo 8

Estado del Arte

El monitoreo de signos vitales ha mostrado mayor notoriedad en las últimas décadas, las cuales apuntan a la implementación de nuevas tecnologías para apoyar diferentes necesidades del sector salud: Implementación de monitoreo a través de sistemas embebidos para cuidado de adultos mayores [46], diseño e implementación de sistema de monitoreo remoto enfocado en la investigación y ejecutado de la mano con el IoT [47], guía de evaluación para mejorar el funcionamiento del monitoreo remoto para pacientes con diabetes [48], etc. Partiendo de que el sistema que se planea desarrollar busca ser una herramienta para mejorar estos procesos, dado que se busca agilidad en procesos internos, versatilidad en el manejo de información que puede ser útil tanto en procesos de monitoreo, como también, en seguimiento de historiales de salud. Para poder realizar un sistema que presente novedad se hace un estudio de proyectos enfocados en sistemas de información.

8.1. An Integrated Healthcare Enterprise Information Portal and Healthcare Information System Framework

El sistema de información en este caso de éxito parte con la premisa de mejorar el desarrollo de procesos dentro del marco hospitalario de una entidad de la salud en Taiwan, también, en el cual se implementa un portal de información empresarial de atención médica integrado y distribuido sobre infraestructura inalámbrica/alámbrica que generó cambios en la cultura del usuario, procesos y paradigmas de como se venía manejando toda la información del centro de salud, dado que centralizan todo el sistema entre la red privada del hospital, que permite una mejora dentro de la institución percibiendo así gasto a nivel de infraestructura y adecuaciones, pero recompensándola con la mejora de procesos. Por último se hace énfasis de que permite mejoras con el cliente dado que se enfocan también en la gestión de relaciones con el cliente (GRC) al implementar en conjunto el Sistema de información hospitalaria (SIH) y un portal de información empresarial de atención médica con acceso single sign-on que permite el inicio de sesión unificado al usar una red propia y una tokenización dentro del sistema [49].

8.2. A review on open source architecture in Geographical Information Systems

El sistema de información expuesto va enfocado a los beneficios a largo plazo y las sugerencias expuestas, de un sistema de información geográfico (SIG) el cuál se crea en 1980 y sigue creciendo hasta generar varios productos de software que lo complementan para su mejora, muestra el desarrollo que ha abarcado el seguir con el uso de software open source que integra diferentes lenguajes de programación para su continua mejora. Se observa que la finalidad almacenar, mapear, analizar

y mostrar datos geográficos de libre acceso para ser usados en varios campos de estudio [50].

8.3. Sistema de Alertas de Signos Vitales y de Ubicación a través de Dispositivos Móviles

El presente proyecto consiste en generar un sistema de alertas móvil enfocado en personas que sufran de arritmias cardíacas con el fin de poder realizar un monitoreo continuo de la frecuencia cardíaca. Estos signos se encuentran alojados en una base de datos en un servidor remoto y por medio de un correo electrónico, el sistema reporta cuando se presenta una anomalía en el sistema cardiaco del paciente; este correo es enviado a los contactos que hayan sido autorizados y al médico de cabecera, junto con la frecuencia cardíaca, la posición GPS y un mapa de la ubicación del usuario. Este proyecto fue posible gracias al uso de dispositivos electrónicos ya existentes para la medición de signos vitales en conjunto con la tecnología de los dispositivos móviles [51].

8.4. Sistema web para el monitoreo de signos vitales y control del historial clínico de pacientes hospitalizados en servicios de medicina interna del hospital general regional "Dr. Rafael Pascasio Gamboa".

Este desarrollo web está diseñado para el área de "Medicina Interna" del hospital Dr. Rafael Pascasio Gamboa, y está enfocado en visualizar los signos vitales provenientes a un módulo Wi-Fi. El sistema de información cuenta con cuatro tipos de usuarios que intervienen directamente con el paciente en cuestión, los cuales son: administrador,

médico (visualiza los signos vitales del paciente, observa su la información clínica, realiza anotaciones y observaciones en formularios especiales, toma decisiones y las dictamina en un formulario), jefa de piso (asigna los pacientes a los médicos y administra la información) y enfermera (registra la información, visualiza los signos vitales de manera local en una tablet y realiza anotaciones y observaciones en formularios especiales), generando la estancia del paciente más eficiente en cada actividad que se tenga que realizar. Además, el sistema genera reportes en formato PDF de todos los pacientes que se encuentren registrados [52].

8.5. Tratamiento de rizados en la estimación de la densidad espectral de potencia de la señal de ritmo cardiaco.

A nivel institucional se pudo observar que la base de datos publica "Physionet.^{es} una ideal herramienta para tomar datos confiables para el análisis de comportamientos reales de pacientes. En este caso, se realiza un procedimientos referente a la detección de variabilidad por medio de la implementación de promediado de espectros en frecuencia cardiaca a partir de ondas electrocardiografías (ECG) tomadas de Physionet.[53]

Capítulo 9

Funcionamiento General del Sistema

El sistema de información busca cumplir tres (3) propósitos principales, que son: visualizar y almacenar datos de signos vitales, y alertar anomalías presentadas en estos signos. Además, existen dos (2) tipos de usuarios que están involucrados:

- Paciente: Usuario capaz de almacenar sus signos vitales en la base de datos interna, visualizar sus datos enviados por el simulador, su historial de signos y de anomalías presentadas.
- Analista: Usuario capaz de visualizar los datos captados por el simulador, historiales de signos vitales y de alertas sobre anomalías presentadas por pacientes.

Para poder cumplir los principales objetivos del sistema de información, se usan diferentes herramientas tecnológicas:

- Visualización: A través de una página web desplegada en Heroku.
- Almacenamiento: Por medio de una base de datos SQL en la nube.
- Alertar: A partir de mensajes por WhatsApp tanto al paciente como al analista, utilizando la herramienta Twilio como servicio de mensajería.

Los signos vitales que ingresan al sistema son tres (3), los cuales cada uno presenta diferentes rangos de anomalías como se puede observar en el cuadro 9.1

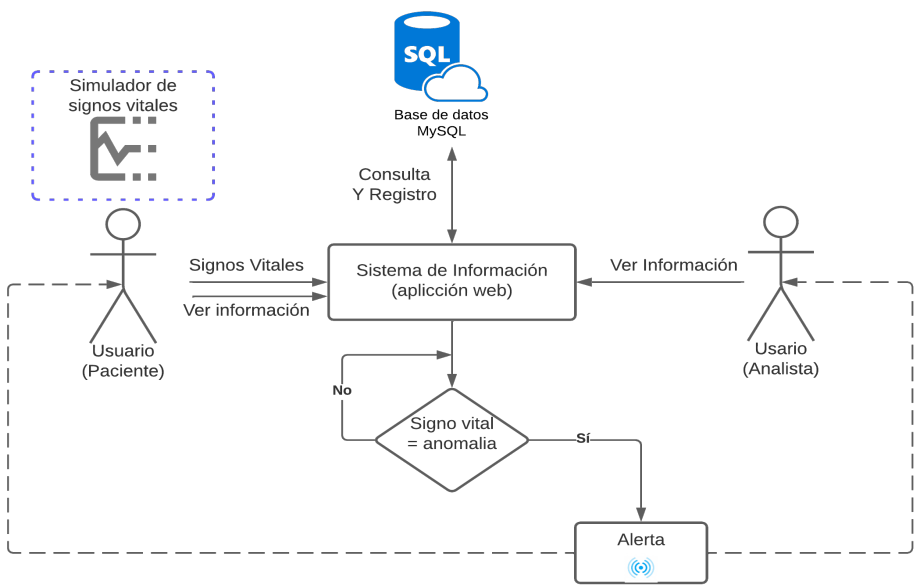
Cuadro 9.1. Rangos de Anomalías según signos vitales

Signo vital	Anomalía	Rango
Frecuencia Cardiaca	Taquicardia	Mayor a 100 latidos por minuto
Frecuencia Cardiaca	Bradicardia	Menor a 60 latidos por minuto
Temperatura	Hipotermia leve	Entre 33°C y 36°C
Temperatura	Hipotermia	Entre 28°C y 32°C
Temperatura	Hipotermia grave	Menor a 28°C
Temperatura	Hipertermia	Mayor 37.5°C
Oximetría	Desaturación	Entre 95 % y 88 %
Oximetría	Grave desaturación	Menor a 87.9 %

Fuente: Propia

Se genera el esquema de la figura 9.1 para explicar con mayor detalle las funcionalidades expuestas dentro del sistema de información en los próximos capítulos. El usuario (Paciente), es remplazado por un simulador alimentado por una base de datos de signos vitales con el fin de simular una herramienta de captación que ingrese información relevante al sistema.

Figura 9.1. Diagrama general del funcionamiento del sistema de información.



Fuente: Propia

Capítulo 10

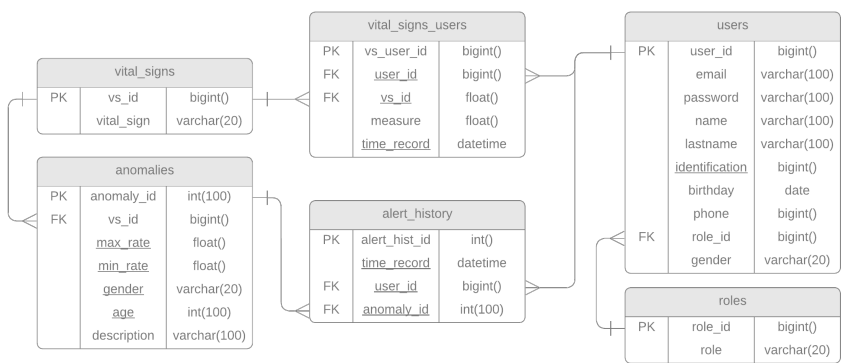
Diseño y creación de Base de Datos

El diseño parte de observar el funcionamiento general que se propuso frente la información de los usuarios con el sistema planteado. Posterior a esto se establece un modelo relacional que busca volver al sistema escalable y que se eviten malas prácticas en el almacenamiento de información SQL.

El modelo relacional observado en la figura 10.1 toma 6 tablas para poder organizar el sistema. En donde el escalamiento del sistema se le atribuye a 3 de estas "vital_signs" ,"anomalies" y "roles". Además, existen 2 de relación "vital_signs_users" y "alert history" para almacenar datos entregados por el simulador y guarda las alertas captadas, en donde referencia las tablas de signos vitales, usuarios y anomalías. Dado que en el sistema de la salud hay muchas más variantes que determinan el estado un paciente, este planteamiento de la base de datos busca que el sistema no se quede con solo 3 signos vitales, sino, que a medida que se presenten las necesidades, más signos vitales puedan ser almacenados, visualizados y generar sus respectivas alertas. Este último, también se

puede generar dado a la habilitación de varios campos provenientes de la tabla 'anomalies' que da plasticidad a qué existan más variaciones en las anomalías por género y edad. Por último, se genera una tabla únicamente en roles para poder generar nuevos campos dependiendo de las necesidades que se desglosen en un futuro y no negando la oportunidad de más tipos de usuarios.

Figura 10.1. Modelo Relacional



Fuente: Propia

Para la creación de las tablas se creó un script para generarlas de manera automática (Ver anexo A), con las llaves foráneas y con los principales valores de la tabla de vital_sings y roles.

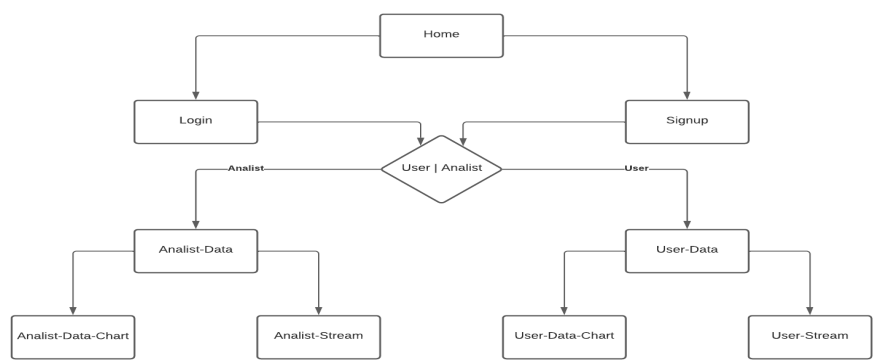
Se usa MySQL como el sistema de datos relacional capaz de conectar con Heroku y que tiene una interfaz (MySQL Workbench) para poder administrar de manera más interactiva las tablas de la base de datos que va a manejar el sistema de información que maneja como estructura SQL.

Capítulo 11

Diseño de Wireframes

El diseño de los wireframes de la página web se realiza con el fin de definir la jerarquía de toda la información que se encuentre allí establecida, los campos necesarios, los botones y los diferentes formularios que la compongan. Antes de comenzar con el diseño, se realiza un mapa web, que muestra cuál es la red de links que tiene la aplicación web y a que ruta redirige cada apartado, como se muestra en la figura 11.1.

Figura 11.1. Mapa Web de la aplicación

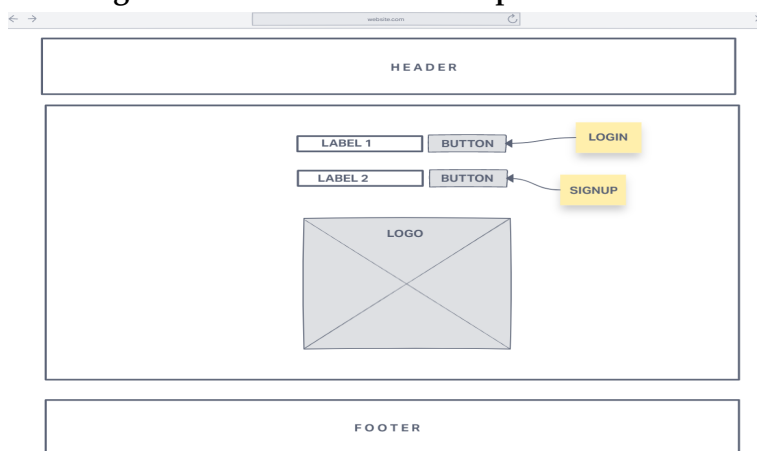


Fuente: Propia

Ya teniendo el mapa web, se proceden a diseñar los diferentes wireframes que tendrá la página web, comenzando por la pestaña inicial que se le muestra al usuario, siguiendo con las demás rutas con las que se cuentan.

- Wireframe Home: es la pestaña inicial que se le muestra al usuario, cuenta con el título de la tesis en la cabecera, en el cuerpo cuenta con dos botones los cuales redirigen a las rutas **/login** y **/signup** para ingresar o registrarse respectivamente, y en el pie de página cuenta con algunos datos de los desarrolladores, este diseño se muestra en la figura 11.2

Figura 11.2. Wireframe de la pestaña Home



Fuente: Propia

- Wireframe Login: esta es una de las rutas de autenticación del usuario, en donde éste ya debe haberse registrado para poder ingresar. El diseño de esta ruta es un formulario en donde el cliente ingresa su identificación junto con su contraseña y un botón para enviar este formulario al servidor como se muestra en la figura 11.3

Figura 11.3. Wireframe de la pestaña Login



Fuente: Propia

- Wireframe Signup: esta ruta también es de autenticación, pero a diferencia de la anterior, esta es para registrarse y guardar a los usuarios en la base de datos. El diseño es un formulario en donde el usuario ingresa datos como nombres, apellidos, cédula, contraseña, fecha de nacimiento, email, celular, el rol que ejercerá ya sea "usuario" o "analista", y el género. El resultado se puede ver en la figura 11.4

Figura 11.4. Wireframe de la pestaña Signup



Fuente: Propia

Dependiendo del rol que elija el cliente, la aplicación redirige a dos rutas principales las cuales son la pestaña del analista o la del usuario, las cuales cuentan con componentes diferentes cada una como se muestra a continuación:

- **Wireframe Analista:** esta ruta, en su cabecera cuenta con el nombre del analista en la esquina superior derecha; y en la esquina superior izquierda hay una barra de navegación la cual cuenta con tres opciones, la primera redirige al historial de los usuarios, la segunda redirige a una ventana donde se muestran los datos en tiempo real, y por último una opción de salir que redirige al inicio. En el cuerpo, hay un menú desplegable con las diferentes opciones de clientes que hayan elegido como rol "usuario", además hay dos campos para ingresar fecha inicial y final para que el analista pueda ver en un rango de tiempo específico el registro de los signos vitales del usuario elegido, por último hay un botón el cual envía los datos mencionados anteriormente y redirige a una nueva ruta (ver figura 11.5).

Figura 11.5. Wireframe del rol Analista

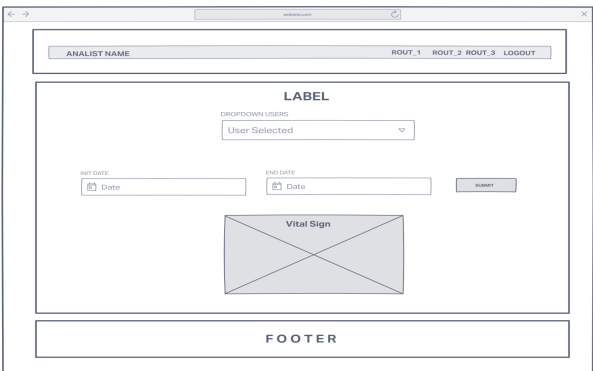
The wireframe illustrates the layout for the Analyst role. It features a top navigation bar with the analyst's name and three route options (ROUT_1, ROUT_2, ROUT_3) along with a logout button. The main body contains a label, a user selection dropdown, date range inputs (start and end dates), and a submit button. A footer is located at the bottom of the page.

Fuente: Propia

En la barra de navegación se cuenta con dos opciones para redirigir a las pestañas de historial de signos vitales por usuario y a la vista en tiempo real de estos. Estos datos se visualizan a partir de gráficas. A continuación, se muestra el diseño de estos dos wireframes

- Wireframe Historial de Datos: dependiendo la fecha inicial y final, y el usuario que elija el analista, en el cuerpo de la página se muestra mediante gráficos, los diferentes datos de los signos vitales por usuario que se hayan registrado en la base de datos como se muestra en la figura 11.6

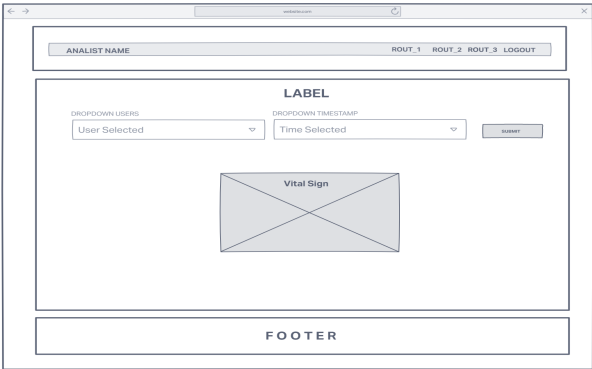
Figura 11.6. Wireframe del Historial de Datos



Fuente: Propia

- Wireframe Datos en vivo: en este caso no es necesario elegir fechas ya que los datos que se muestran acá son consultas en tiempo real. En el cuerpo de la página se encuentran dos menús desplegables, el primero para elegir el usuario y el segundo para elegir cuantos datos se quieren visualizar en los gráficos como se muestra a continuación en la figura 11.7

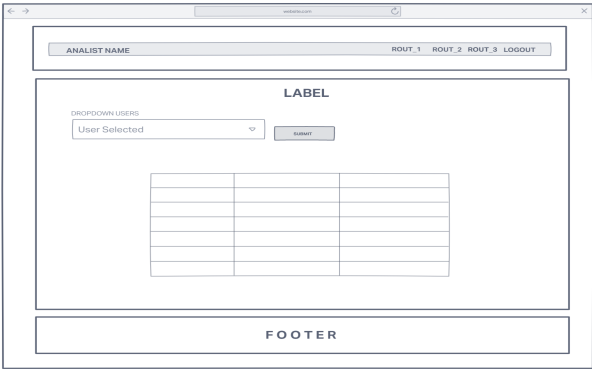
Figura 11.7. Wireframe Datos en vivo



Fuente: Propia

- Wireframe Historial de Datos: se cuenta con un menú desplegable para poder elegir el usuario al cual se le quieren consultar el historial de alertas, y un botón para realizar la consulta a la base de datos; la información se muestra a través de una tabla, como se muestra en la figura 11.8

Figura 11.8. Wireframe Historial de Alertas



Fuente: Propia

- Wireframe Usuario: al igual que en la pestaña del analista, en la cabecera cuenta con el nombre del usuario en la parte superior derecha y una barra de navegación en la parte superior izquierda. En el cuerpo de la página, está la identificación de dicho usuario junto con dos campos para ingresar fecha inicial y final para que el usuario pueda observar en el rango de tiempo elegido las gráficas de sus signos vitales como se muestra en la figura 11.9

Figura 11.9. Wireframe del rol Usuario

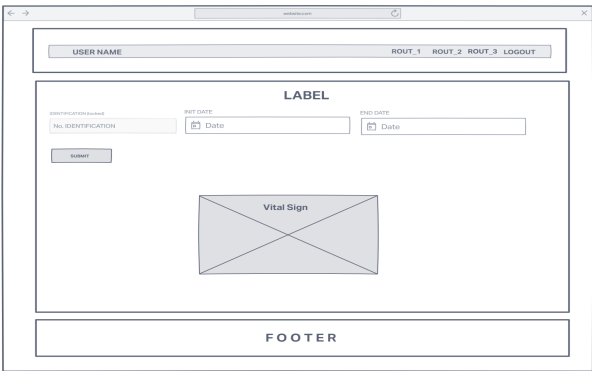


Fuente: Propia

En este caso la barra de navegación también redirige a dos pestañas las cuales son el historial de signos vitales del usuario y la pestaña de visualización en tiempo real de estos mismos. Estos datos se visualizan a partir de gráficas. A continuación, se muestra el diseño de estos dos wireframes

- Wireframe Historial de Datos: dependiendo la fecha inicial y final en el cuerpo de la página se muestra mediante gráficos, los diferentes datos de los signos vitales de dicho usuario que se hayan registrado en la base de datos como se muestra en la figura 11.10

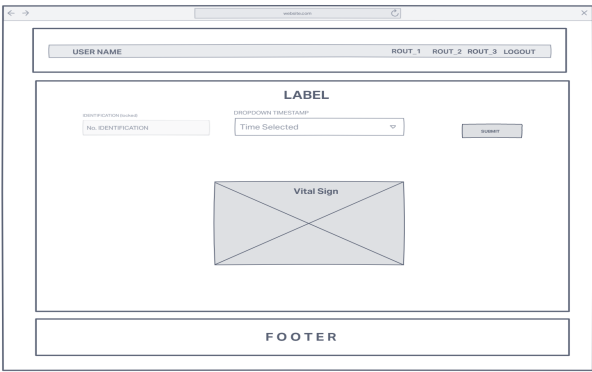
Figura 11.10. Wireframe del Historial de Datos



Fuente: Propia

- Wireframe Datos en vivo: en este caso no es necesario elegir fechas ya que los datos que se muestran acá son consultas en tiempo real. En el cuerpo se encuentra la identificación del usuario que está actualmente con la sesión abierta y además hay un menú desplegable para elegir cuantos datos se quieren visualizar en los gráficos como se muestra a continuación en la figura 11.11

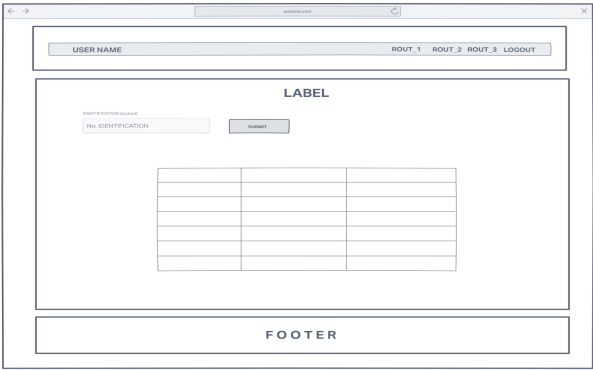
Figura 11.11. Wireframe Datos en vivo



Fuente: Propia

- Wireframe Historial de Alertas: este diseño cuenta solamente con un botón el cual trae la información del historial de alertas del usuario, y las muestra por medio de una tabla, como se muestra en la figura 11.12

Figura 11.12. Wireframe Historial de Alarmas



Fuente: Propia

Capítulo 12

Estructura del Back- End

Para desarrollar el servidor de la aplicación web, se ha decidido implementarlo en el lenguaje JavaScript. Este lenguaje actualmente cuenta con cantidad de bibliotecas y frameworks disponibles para el desarrollo web, además es uno de los lenguajes de programación que más se usa actualmente en la comunidad que desarrolla back-end, front-end o full stack [54].

Implementar el servidor desde cero puede resultar muy tedioso y más complejo, pero al usar JavaScript, se tiene la posibilidad de desarrollar el back-end con un entorno multiplataforma de código abierto como lo es Node.js, basado en este mismo lenguaje.

Además de brindar un entorno multiplataforma de código abierto, Node.js da la facilidad de realizar desarrollos de back-end con un framework llamado Express.js que simplemente es una herramienta que ayuda a que la implementación del servidor sea más sencilla a la hora codificar.

12.1. Estructura del servidor

Para descargar e instalar Node.js es necesario dirigirse a la página oficial y desde allí seleccionar el archivo, dependiendo de su sistema operativo. También es necesario descargar el gestor de paquetes de este mismo, que se llama **npm** con el cual se pueden instalar dependencias y paquetes que se requieran, entre ellos el framework Express. Para instalar las dependencias que se necesiten para el programa, se instalan usando el gestor de paquetes mencionado anteriormente, digitando las siguientes líneas en la terminal.

```
~$ npm init
```

```
~$ npm install dependencies
```

Con el comando **npm init** se genera un archivo *package.json* en donde se encuentra la información general del proyecto, el nombre y las dependencias y con el comando **npm install** se pueden instalar las diferentes dependencias que se necesiten para el desarrollo del programa, como es el caso de Express. A continuación, se puede observar en el código 12.1 con las dependencias que se van a usar.

Código 12.1. Archivo .json del programa

```
1 {  
2   "name": "sistema-informacion",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "index.js",  
6   "scripts": {  
7     "test": "echo \"Error: no test specified\" && exit 1",  
8     "dev": "nodemon src/",  
9     "start": "node src/index.js"  
10  },  
11  "keywords": [],  
12  "author": "",  
13  "license": "ISC",  
14  "dependencies": {
```

```
15   "bcryptjs": "^2.4.3",
16   "connect-flash": "^0.1.1",
17   "dotenv": "^16.0.0",
18   "express": "^4.17.3",
19   "express-handlebars": "^6.0.3",
20   "express-mysql-session": "^2.1.7",
21   "express-session": "^1.17.2",
22   "express-validator": "^6.14.0",
23   "moment": "^2.29.2",
24   "morgan": "^1.10.0",
25   "mysql": "^2.18.1",
26   "passport": "^0.5.2",
27   "passport-local": "^1.0.0",
28   "twilio": "^3.76.1"
29 },
30 "devDependencies": {
31   "nodemon": "^2.0.15"
32 }
33 }
```

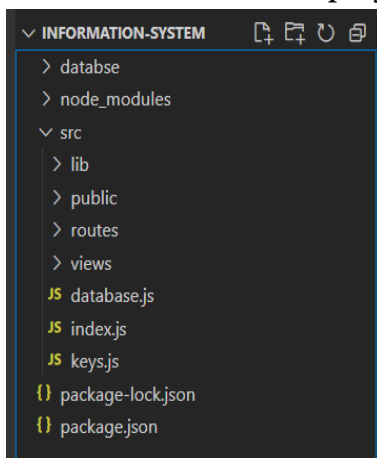
En la sección de las dependencias se puede evidenciar la instalación de las librerías que se van a utilizar en el proyecto, a continuación se realiza una breve descripción de cada una de ellas y su funcionalidad en el proyecto:

- **bcryptjs**: es una función de hashing que ayuda a cifrar las contraseñas, y en el proyecto es utilizada con ese fin.
- **connect-flash**: este módulo se usa especialmente para enviarle mensajes por pantalla cada vez que el usuario se redirige a una ruta en específico dentro de la aplicación web.
- **express**: es el framework sobre el cual se va a desarrollar el servidor de la aplicación y con este módulo se configura el puerto, los middlewares y demás configuraciones que necesite el servidor.
- **express-handlebars**: este módulo ayuda a relacionar el motor de vistas Handlebars con el framework Express para poder comunicar fácilmente el servidor con las vistas del front-end.

- `express-mysql-session`: la dependencia crea un grupo de conexiones interno de MySQL guardado en una sesión que se encarga de establecer la conexión entre el programa y la base de datos con ayuda de las llaves o credenciales que esta tenga, como lo son el usuario, la contraseña, el nombre del host y de la base de datos.
- `express-session`: es un middleware encargado de almacenar la identificación de los datos de sesión del servidor en una cookie.
- `express-validator`: el módulo es un conjunto de middlewares encargados de validar la información de la sesión creada.
- `moment`: es una librería utilizada para formatear fechas dependiendo de lo que se le quiera mostrar al usuario.
- `morgan`: es un middleware de solicitud de HTTP la cual registra todas las solicitudes que se hagan en la aplicación web.
- `mysql`: la dependencia es un driver para MySQL el cual está escrito en lenguaje Javascript el cual no requiere compilación.
- `passport`: sirve para gestionar la autenticación en el registro e ingreso de los usuarios a la página y lo realiza a través de un conjunto extensible de complementos conocidos como estrategias.
- `passport-local`: es la estrategia elegida de la dependencia `passport` para que el usuario se autentique con nombre de usuario y contraseña.
- `twilio`: esta dependencia sirve para relacionar el programa con el servicio de mensajería de Twilio para enviarle al usuario mensajes vía Whatsapp cuando se presente una anomalía en sus signos vitales.

La estructura del proyecto está dividida por secciones, las cuales ayudan a organizar el código. Al estar dividido por carpetas, es posible saber cuáles archivos están dentro y cuál es su función. La estructura principal del proyecto se muestra a continuación en la figura 12.1.

Figura 12.1. Estructura del proyecto



Fuente: Propia

Esta es la estructura principal del proyecto. Todo se encuentra en la carpeta llamada **SISTEMA-INFORMACION**, allí se encuentra la carpeta principal del proyecto **src** en donde están las demás carpetas y el archivo *index.js*, en el cual se encuentran todas las dependencias, middlewares, variables globales, funciones y demás que hacen que funcione correctamente la página web.

El archivo *keys.js* contiene las llaves de username, password, hostname y database que fueron generados cuando fue creada la base de datos en la plataforma Heroku; y en el archivo *database.js* se establece la conexión entre el programa y la base de datos alojada en Heroku, esto se hace gracias a la dependencia instalada con npm de MySQL.

En la carpeta **src** hay diferentes subcarpetas, las cuales contienen archivos que serán de ayuda para el desarrollo de la página:

- **Carpeta lib:** se encuentran los archivos que ayudan a la autenticación del usuario en su registro e inicio de sesión, además se encuentra un archivo en el cual se encripta la contraseña del usuario para que le de seguridad a la aplicación.
- **Carpeta public:** allí se alojan dos tipos de archivos, los primeros son archivos CSS los cuales son los encargados de darle estilo a la página; y los archivos como imágenes que se encuentran en la página.
- **Carpeta routes:** esta es una de las carpetas más importantes del proyecto, ya que ahí se encuentran las rutas de la página web, se especifica si la ruta tiene un método **GET** o un método **POST** y en las mismas rutas, se renderiza lo que se va a mostrar al usuario en la parte del front-end. Además de estos endpoints de visualización, se crea una ruta para captar la información proveniente del simulador de señales para poderla guardar en la base de datos.
- **Carpeta views:** al igual que la anterior carpeta, esta es una de las más importantes, ya que en esta se encuentran todos los archivos de front-end. Esto se hace con la ayuda de *Handlebars* que es una librería de JavaScript, es un popular sistema de plantillas que permite crear y formatear código en HTML, siendo más sencillo alterar el DOM.

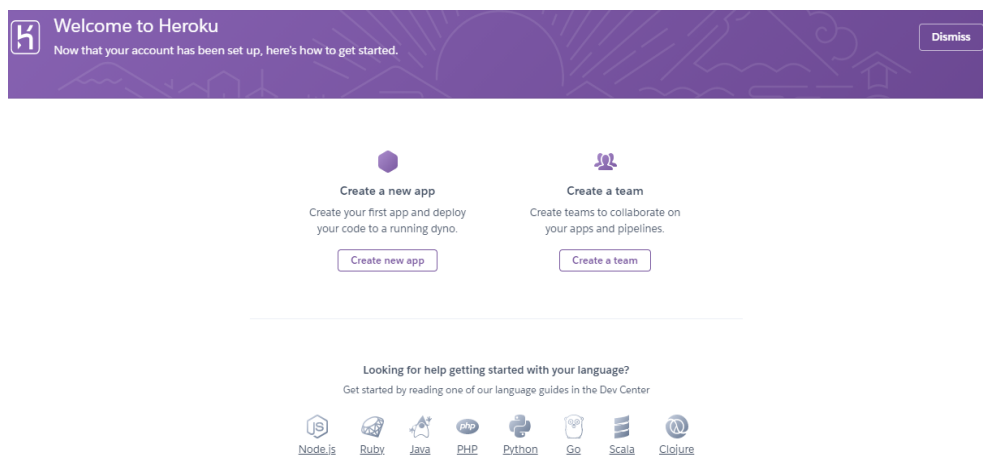
12.2. Conexión entre Heroku y MySQL

Para el registro en esta plataforma es necesario seguir los siguientes pasos:

- Registrarse en la página oficial de Heroku (<https://signup.heroku.com>), dado que es obligatorio para utilizar la aplicación. Llegará un correo de confirmación para generar una nueva contraseña y poder acceder a la plataforma.

- Aceptar términos de servicio de la plataforma Heroku, para poder continuar con el proceso de registro.
- Una vez hechos estos pasos esta es la pestaña de Inicio de Heroku, en donde proporcionan varias guías de desarrollo para los diferentes lenguajes de programación que soporta la plataforma (ver figura 12.2).

Figura 12.2. Ventana de Inicio de la plataforma Heroku



Fuente: <https://dashboard.heroku.com/apps>

Para realizar la conexión entre la plataforma Heroku y la base de datos que se va a utilizar, en este caso es MySQL, Heroku proporciona una serie de complementos llamados Add-ons que básicamente son herramientas, las cuales sirven para ampliar el desarrollo de las aplicaciones en esta plataforma.

A continuación, se muestran una serie de pasos a seguir para establecer la conexión entre la plataforma y la base de datos:

- Descargar MySQL Workbench, para el desarrollo en SQL y administración en la configuración del servidor, de usuarios, etc. de manera integral. Otra opción si se está trabajando en un sistema

operativo diferente a Windows, como Linux, MySQL se puede trabajar desde la terminal del sistema

- Crear una nueva aplicación en Heroku con un nombre adecuado para sus fines, este paso se puede realizar por medio de la plataforma, o instalando Heroku CLI para crearla desde la terminal. Para crear la aplicación desde terminal, es necesario instalar la dependencia Heroku CLI; esta descarga dependerá del sistema operativo en el cual se encuentre la persona. Una vez descargado, desde terminal se puede crear la aplicación de la siguiente forma:

- `~$ heroku login`
- `~$ heroku create [nombre de la app]`

- Luego de tener la aplicación creada en Heroku, se procede a realizar la conexión entre MySQL y Heroku, para esto, con la ayuda de Heroku CLI, en la terminal se escribe la siguiente línea de comandos

- `~$ heroku addons:create cleardb:ignite`

- Una vez hecha la instalación de ClearDB en la aplicación, es necesario obtener la URL de la base de datos para obtener las credenciales de esta y así más adelante poder establecer la conexión entre el programa y la base de datos asociada a la aplicación de Heroku; con el siguiente comando se obtiene la URL

- `~$ heroku config | grep CLEARDB_DATABASE_URL`

- El comando anterior arrojará una URL con información importante para configurar la base de datos de MySQL, la URL será parecida a la siguiente línea:

- `mysql://b041adab13d426:1c9a23a3@us-cdbr-east-05.cleardb.net/heroku_a72350352312ba8?reconnect=true`

En esta línea están los datos importantes como el nombre y la contraseña de la base de datos y el host, pero primero hay que copiar toda la URL e introducir el siguiente comando en la terminal

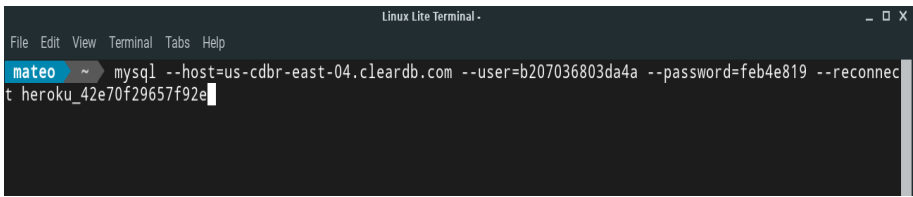
```
~$ heroku config:set DATABASE_URL='mysql://b041adab  
13d426:1c9a23a3@us-cdbr-east-05.cleardb.net/heroku  
_a72350352312ba8?reconnect=true'
```

- Con esto ya se tendrá configurada la base de datos en la aplicación de Heroku, y a continuación se muestran cuáles son los datos del nombre de usuario, la contraseña, el host de la base de datos y el nombre de esta, que resultan a partir de la URL mostrada anteriormente.
 - mysql:// (información no relevante)
 - b041adab13d426 (**USERNAME**)
 - : (información no relevante)
 - 1c9a23a3 (**PASSWORD**)
 - us-cdbr-east-05.cleardb.net (**HOSTNAME**)
 - / (información no relevante)
 - heroku_a72350352312ba8 (**DATABASE NAME**)
 - ?reconnect=true (información no relevante)

Con esta serie de pasos ya quedó configurada y asociada la base de datos de MySQL con el Add-on ClearDB en la plataforma Heroku. Gracias a esto, en el desarrollo del código de la página web se puede establecer la conexión con esta base de datos para guardar los distintos registros en las tablas que sean creadas en esta misma.

Para acceder desde la terminal a la base de datos, es necesario la anterior configuración entre Heroku y MySQL con ClearDB, para haber generado la URL de la base de datos y con esto obtener el nombre del usuario, la contraseña, el host y el nombre de la base de datos, y en la terminal se ingresan estos datos como se muestra en la imagen 12.3

Figura 12.3. Acceso a MySQL desde la terminal de Linux



Fuente: Propia

Con este comando en la terminal se está accediendo a la base de datos que está alojada en Heroku, la cual va a tener tablas una vez el scrpit nombrado anteriormente (Ver anexo A) se ejecute ya sea en MySQL Workbench o en la terminal de Linux. Esto da como resultado la base de datos planteada en la figura 10.1 pero esta vez en la nube.

Capítulo 13

Estructura del Front-End

Para realizar la parte visual de la página, con la que el cliente va a interactuar, se desarrolla por medio de HTML y CSS con la ayuda de Handlebars el cual es un sistema de plantillas en JavaScript que permite crear y formatear código hecho en HTML con el fin de crear secciones en las cuales se pueda traer información del lado del servidor y mostrarla de una manera sencilla en pantalla [55].

Además de utilizar Handlebars, se implementa un framework de CSS llamado Bootstrap que está diseñado para aplicaciones de front-end, el cual proporciona herramientas ya creadas que estilizan los elementos HTML de la página web fácilmente y sin necesidad de crear estilos propios de CSS [56].

13.1. Parte visual de la aplicación web

Una vez terminado el código, ya es posible tener una parte visual del sitio web, a continuación, se pueden apreciar una serie de imágenes mostrando las pestañas de la página

- **Pestaña principal del sitio Web:** esta se observa una vez se abre el link en cualquier navegador, en esta pestaña se encuentran dos botones, uno para iniciar sesión, y el otro para poder registrarse (ver figura 13.1

Figura 13.1. Pestaña inicial



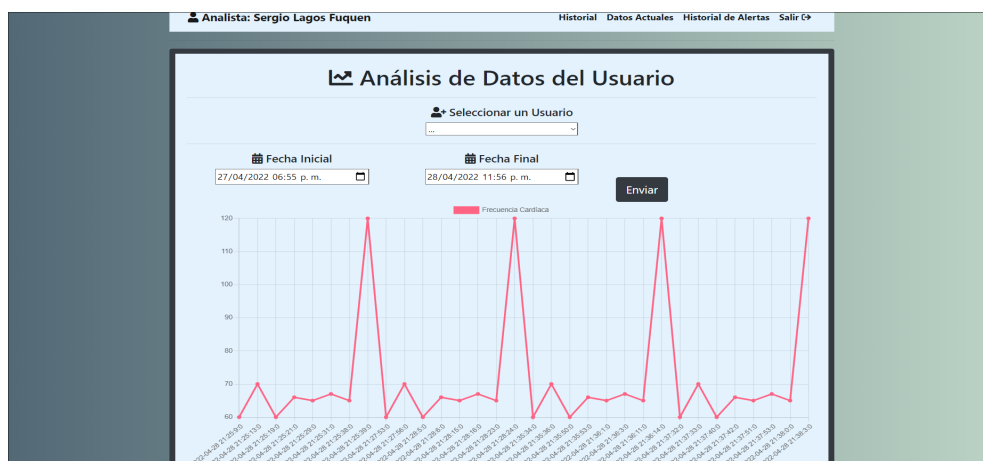
Fuente: Propia

Como se mencionó en el diseño de los wireframes, a la hora de registrarse, el cliente tiene la posibilidad de elegir qué función desempeña, si es Usuario o Analista, y dependiendo del rol que haya escogido se visualizan las ventanas relacionadas, en donde el usuario podrá ver datos en tiempo real, el historial de datos tomados y el historial de alertas cuando se haya presentado alguna anomalía en sus signos vitales.

13.1.1. Cliente con rol de Analista

- **Pestaña del Historial de Datos del Usuario:** permite al Analista seleccionar a los clientes que tengan como rol Usuario y establecer una fecha de inicio y fin para visualizar los datos en ese rango de tiempo. Una vez seleccionado el intervalo de tiempo y el usuario, por medio de gráficas, se ven los datos de los signos vitales, como se puede ver a continuación (ver figura 13.2)

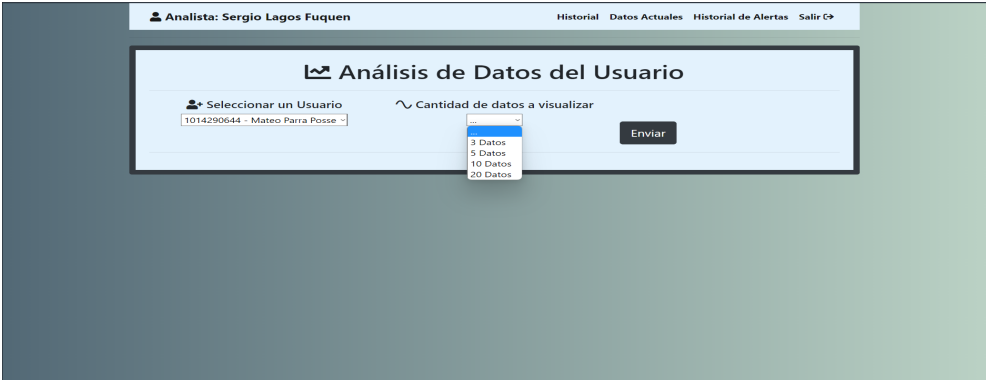
Figura 13.2. Pestaña del historial de datos del usuario



Fuente: Propia

- **Pestaña de Datos en vivo:** en esta ventana el Analista tiene la opción de seleccionar la cantidad de datos que quiere observar en la gráfica en tiempo real, además de poder seleccionar al Usuario del cual quiera ver los signos vitales como se muestra en la figura 13.3.

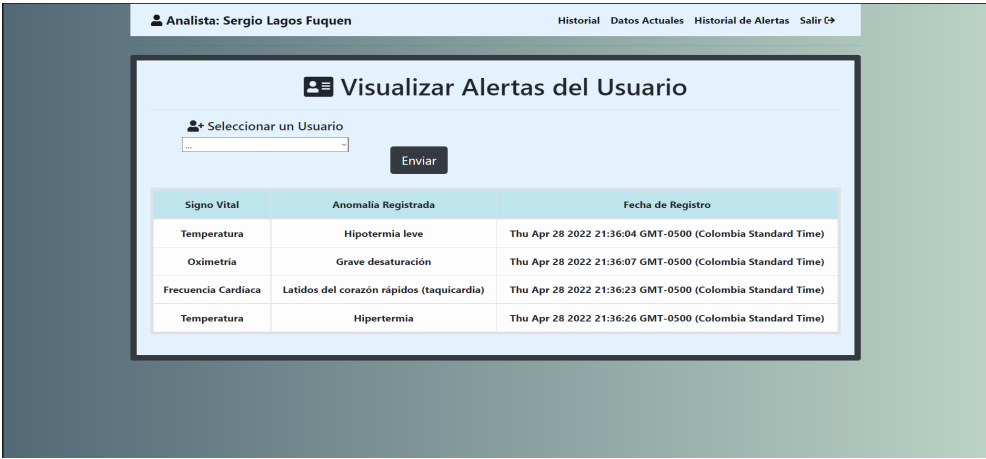
Figura 13.3. Pestaña de visualización de datos en vivo del usuario



Fuente: Propia

- **Pestaña Historial de Alertas:** esta pestaña le da la posibilidad al Analista de elegir el usuario y ver mediante una tabla el historial de alertas que se han generado cuando ha existido algún tipo de anomalía en los signos vitales, la tabla muestra el signo vital relacionado, la descripción de la anomalía y la fecha en la cual se presentó (ver figura 13.4).

Figura 13.4. Pestaña de historial de alertas del usuario

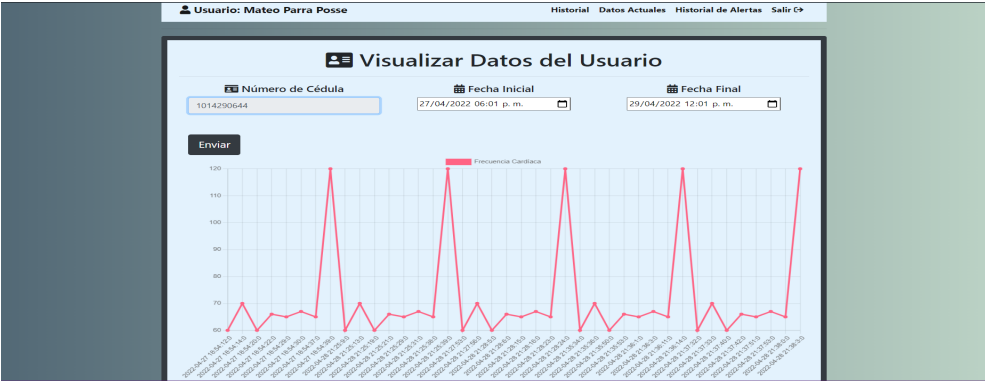


Fuente: Propia

13.1.2. Cliente con rol de Usuario

- **Pestaña del Historial de Datos:** permite al Usuario establecer una fecha de inicio y final para visualizar sus signos vitales por medio de gráficas, como se puede ver a continuación (ver figura 13.5)

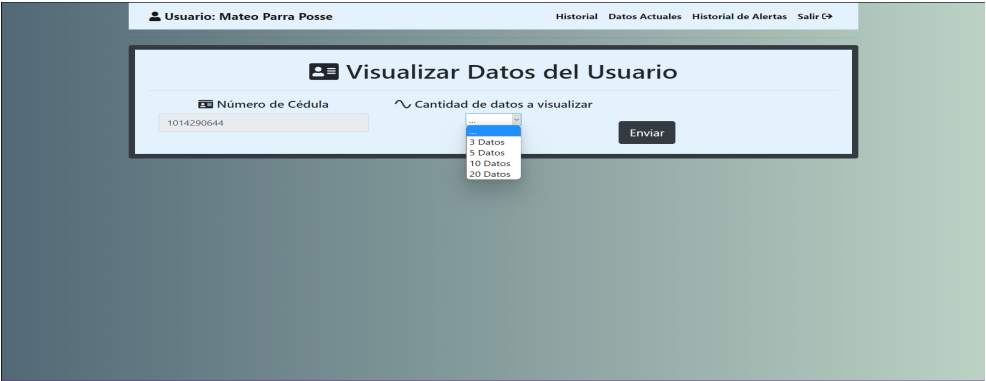
Figura 13.5. Pestaña del historial de datos



Fuente: Propia

- **Pestaña de Datos en vivo:** en esta ventana el Usuario tiene la opción de seleccionar la cantidad de datos que quiere observar en la gráfica en tiempo real, como se muestra en la figura 13.6.

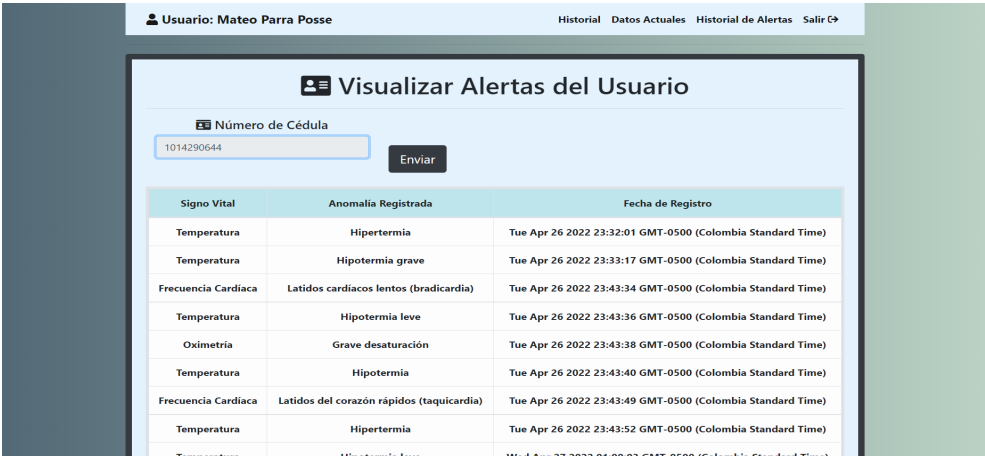
Figura 13.6. Pestaña de visualización de datos en vivo



Fuente: Propia

- **Pestaña Historial de Alertas:** esta pestaña le da la posibilidad al Usuario ver mediante una tabla el historial de alertas que se han generado cuando ha existido algún tipo de anomalía en sus signos vitales, la tabla muestra el signo vital relacionado, la descripción de la anomalía y la fecha en la cual se presentó (ver figura 13.7).

Figura 13.7. Pestaña historial de alertas



Fuente: Propia

Capítulo 14

Generación de Alarmas

La finalidad de la generación de alarmas es alertar al Usuario y a los Analistas cuando se presente algún tipo de anomalía en sus signos vitales. Las anomalías se toman a partir de la investigación hecha en el estado del arte y se registran en la base de datos en tabla llamada 'anomalies' (ver figura 14.1), con el fin de que el sistema de información sea escalable en un futuro y se puedan insertar más tipos de anomalías dependiendo la necesidad que se pueda llegar a requerir.

Figura 14.1. Tabla de Anomalías tomada de la base de datos

anomaly_id	max_rate	min_rate	description	vital_sign
1	200	100.1	Latidos del corazón rápidos (taquicardia)	Frecuencia Cardíaca
2	59.9	0	Latidos cardíacos lentos (bradicardia)	Frecuencia Cardíaca
3	95	88	Desaturación	Oximetría
4	87.9	0	Grave desaturación	Oximetría
5	36	33	Hipotermia leve	Temperatura
6	32.9	28	Hipotermia	Temperatura
7	27.9	0	Hipotermia grave	Temperatura
8	60	37.5	Hipertermia	Temperatura

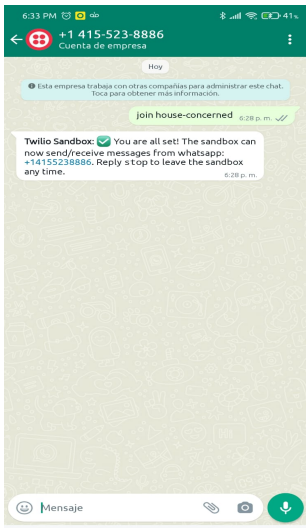
Fuente: Propia

14.1. Plataforma Twilio

Estas alertas se generan por medio de Twilio, una plataforma de desarrollo que permite construir aplicaciones de comunicación en la nube por medio de API's. Para este proyecto las alertas le llegarán al usuario por medio de mensajería, específicamente por mensajes de Whatsapp, ya que un estudio global (Consumer Engagement Best Practices) revela que si los usuarios requieren una comunicación más urgente prefieren el aviso por medio de un mensaje de texto o vía WhatsApp [57], y la plataforma Twilio cuenta con este servicio.

Para registrarse en la plataforma es necesario dirigirse a la página oficial de Twilio y crear una cuenta. Una vez ingresado, hay que realizar la configuración para comenzar a recibir los mensajes del bot, que es un número de Estados Unidos el cual es el siguiente: (+1) 415 523 8886 y se debe enviar el mensaje 'join house-concerned' para asociarlo con el número celular del Analista o del Usuario como se muestra en la figura 14.2

Figura 14.2. Mensaje de confirmación plataforma Twilio

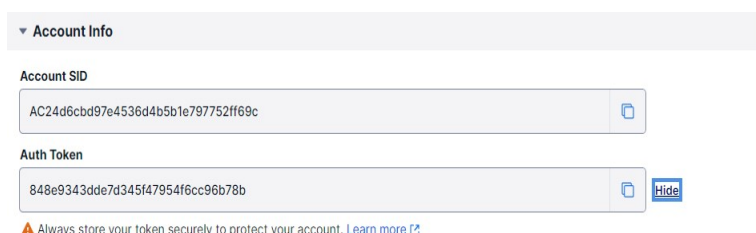


Fuente: Propia

14.2. Mensaje de Alerta desde el código

Una vez se haya vinculado el número celular, en la consola de Twilio, se encuentran las credenciales "Account SID" y "Auth Token" (ver figura 14.3) que permiten la conexión para enviar los mensajes desde el código principal de la aplicación.

Figura 14.3. Credenciales de Twilio



Fuente: Propia

Ya con estas credenciales, se desarrolla una sección en el programa, encargada de identificar las anomalías según los rangos establecidos en la base de datos y de tomar los números de los Analistas y el Usuario para enviar los respectivos mensajes (ver Anexo B) mediante consultas a la base de datos.

Capítulo 15

Despliegue de la aplicación en Heroku

Para poder realizar el despliegue es necesario tener instalado el CLI de Heroku, ya que para hacer cambios y poder subirlos a la nube, se necesita esta herramienta además de tener Git¹ instalado. Inicialmente es necesario iniciar sesión con Heroku CLI.

- `~$ heroku login`

Luego es necesario clonar el repositorio del proyecto para así tener los últimos cambios hechos y poder realizar los nuevos.

- `~$ heroku git:clone -a vital-sings-app`
- `~$ cd vital-sings-app`

¹Es un sistema de control de versiones distribuido usado como repositorio de control de versiones.[58]

Una vez que se ha clonado el repositorio y dirigido a la carpeta del proyecto, se puede usar cualquier editor de código para realizar los cambios y posteriormente hacer el despliegue de la aplicación usando la herramienta Git, escribiendo los siguientes comandos en el terminal.

- `~$ git add .`
- `~$ git commit -m "make it better"`
- `~$ git push heroku master`

Capítulo 16

Procesamiento Base de Datos Physionet

PhysioNet es una página web encargada de permitir el acceso a bases de datos públicas obtenidas de investigaciones, para señales fisiológicas complejas, la cual está respaldada por el Laboratorio de Fisiología Computacional del MIT y el Centro Médico Beth Israel Deaconess [59].

Para la finalidad de este proyecto se toman las siguientes Bases de datos para alimentar el simulador de señales y poder tener registros de usuarios reales:

1. Non-EEG Dataset for Assessment of Neurological Status

Provenientes de una publicación de IEEE [60], esta base de datos contiene señales fisiológicas no EEG para la evaluación del estado neurológico, recopiladas en el Laboratorio de Calidad de Vida de la Universidad de Texas en Dallas. Los datos recopilados consisten en actividad electro dérmica, temperatura, aceleración, frecuencia cardíaca y nivel de oxígeno arterial [61]

2. Treadmill Maximal Exercise Tests from the Exercise Physiology and Human Performance Lab of the University of Malaga.

Proveniente de un artículo en Research in Sports Medicine An International Journal"[62]. Es un conjunto de las medidas cardiorrespiratorias adquiridas durante 992 pruebas de ejercicio máximo graduado (GET) en cinta ergo métrica realizadas en el Laboratorio de Fisiología del Ejercicio y Rendimiento Humano de la Universidad de Málaga. Los datos recopilados consisten en frecuencia cardíaca, el consumo de oxígeno, la generación de dióxido de carbono y la ventilación pulmonar se miden respiración a respiración junto con la velocidad de la cinta rodante durante las pruebas de esfuerzo máximo[63].

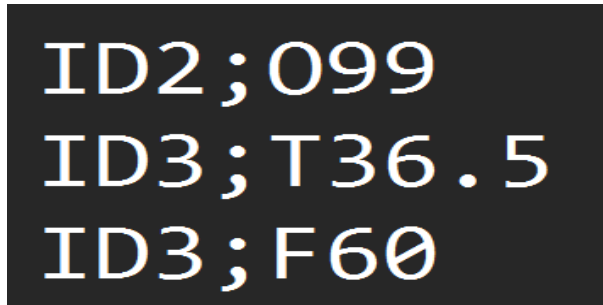
3. Pattern Analysis of Oxygen Saturation Variability

Proveniente del análisis de patrones de la variabilidad de la saturación de oxígeno en individuos sanos en la universidad de Kentucky [64]. Esta base de datos contiene mediciones de saturación de oxígeno de una hora de 36 pacientes [65].

Dado que las bases de datos poseen diferentes formatos de guardado y variaciones en su escritura, se decide procesar cada una de estas con el fin de generar un programa, que simule la captación de la información de diferentes pacientes, la cual es transmitida al sistema de información.

Para organizarlo se decide generar un formato que normalice la entrega de los datos al simulador; para esto como se muestra en la figura 16.1, la información se organizara línea a línea de a un paciente con un signo vital en específico. La primer información leída de la línea va a ser el ID del usuario, seguido de ";" para separarlo de la medida tomada, que comienza con la primera letra del signo vital seguido del valor captado.

Figura 16.1. Ejemplo de formato de normalización de Signos Vitales



```
ID2;099
ID3;T36.5
ID3;F60
```

Fuente: Propia

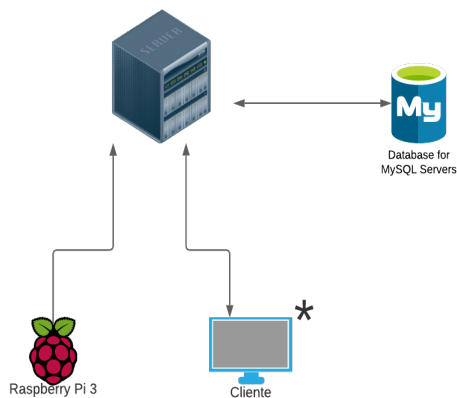
Esta normalización de formato, permite la lectura de manera rápida y simple para posteriormente enviar los datos hacia el sistema de información.

Capítulo 17

Construcción Simulador

El objetivo del simulador de información de signos vitales es alimentar al sistema con datos reales captados de uno o varios dispositivos de obtención de bioseñales para ser almacenados, visualizarlos y generar las alarmas cuando se presenten anomalías en los datos recibidos (Ver figura 17.1).

Figura 17.1. Esquema general del simulador con el sistema.



Fuente: Propia

El proceso de construcción va enfocado en 2 partes principales:

- **Desarrollo de solución de consola:** Esta etapa se enfoca en la forma en que se va a ingresar la información desde el programa (alimentado por una base de datos de signos vitales) hacia el sistema de información.
- **Integración de la solución a un hardware:** Esta etapa se centra en la configuración necesaria, para poder ejecutar el programa creado, en un dispositivo versátil como lo es la Raspberry Pi 3.

17.1. Creación de programa en C#

Se parte de la idea de que la aplicación debe ser en un lenguaje versátil y que permita el manejo de aplicaciones web. Se opta por una solución de consola .NET, la cual maneja lenguaje C#, dado que debe ser ejecutado desde consola en el hardware y este se debe comunicar con aplicaciones web sin problemas.

Para la creación del programa se tuvieron en cuenta 2 etapas en especial:

17.1.1. Envío de información a la página web:

Para esta etapa se tuvo en cuenta el envío de información a la página web; tal como se comentó en el capítulo de la estructura del back-end, se asocia una ruta para el ingreso de la información llamado "information-data" a través del envío de parámetros a través de la solicitud HTTP.

Para poder ingresar algún valor de los parámetros de entrada que

tiene esta ruta, se debía poner seguido a la ruta especificada, "*?nombre del parámetro=valor del parámetro*", para poder continuar ingresando parámetros se pone el carácter "&" y el mismo formato indicado arriba. Un ejemplo de este ingreso de datos es:

```
https://information-system-project.herokuapp.com/  
information-data?user_id=4&vs_id=1&measure=50
```

Los únicos parámetros que tiene en esta ruta como entrada son:

- **user_id** Ingresa el número ID del usuario de quien se está tomando los datos.
- **vs_id** Ingresa el número ID del signo vital que se está tomando.
- **measure** Medida captada por la herramienta.

Una vez teniendo clara la forma de ingreso desde el simulador, se usa una clase para enviar solicitudes HTTP (HttpClient) que mediante el método "GetAsync" envía una solicitud GET al identificador de recursos uniforme (URI) especificado como una operación asíncrona. Esta solicitud enviada, es el formato con los valores modificados que va a leer el programa.

17.1.2. Lectura de archivo txt, con la información

Se usa la clase StreamReader con el propósito de hacer la lectura línea por línea del documento txt que contiene la información procesada de Physionet. Mediante el método ReadLine() se capta línea por línea la información para ser procesada y determinar los parámetros necesarios para enviarlos mediante la solicitud HTTP.

Por último en la creación del código se genera el proyecto, uniendo las 2 etapas anteriores y se unen en un while que pare en el momento que

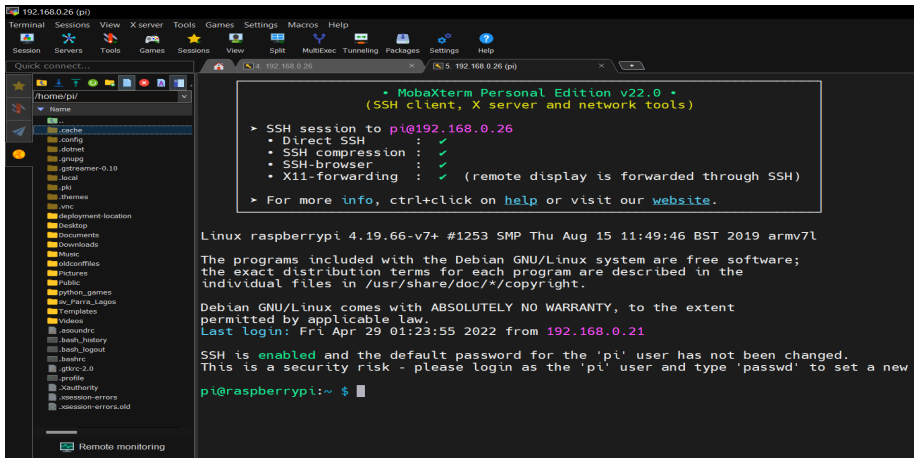
no encuentre más líneas en el documento txt. (El código final se puede observar en el anexo C)

17.2. Integración solución con Hardware

17.2.1. Configuración Tarjeta Raspberry Pi 3

Para el funcionamiento de la Raspberry Pi 3 se inserta una tarjeta Micro SD con sistema operativo (SO) Raspberry Pi instalado, se conecta a la red para actualizar los paquetes disponibles y sus versiones en los repositorios que vienen por defecto en el sistema operativo y se habilita el SSH para poder manipularse desde un servidor en la misma red. Una vez se haya realizado el proceso inicial para el manejo de este hardware de manera remota se usa el software MobaXterm para poder a ingresar tanto a su terminal como a sus carpetas de manera remota (ver figura 17.2), únicamente conociendo la ip que tenía la Raspberry en la red y las credenciales de ingreso para acceder por SSH las cuales venían por defecto (usuario: pi ; contraseña: raspberry)

Figura 17.2. Interfaz Mobaxterm de la Raspberry



Fuente: Propia

Para que el sistema operativo Raspberry Pi acepte la compilación de programas generados en C#, es necesario la instalación de .NET en el dispositivo con los scripts de dotnet-install [66], para esto, se usan las siguientes líneas de código en la terminal, que descargan dotnet y crea una variable de entorno para la ejecución de los programas de C#. (Ver código 17.1)

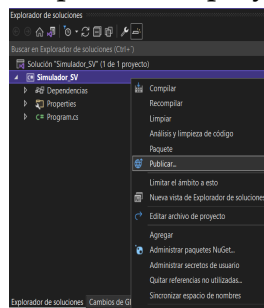
Código 17.1. Comandos de terminal para poder ejecutar .NET en la Raspberry Pi 3

```
1 {  
2 curl -sSL https://dot.net/v1/dotnet-install.sh | bash /dev/stdin  
   --channel Current  
3 echo 'export DOTNET_ROOT=$HOME/.dotnet' >> ~/.bashrc  
4 echo 'export PATH=$PATH:$HOME/.dotnet' >> ~/.bashrc  
5 source ~/.bashrc  
6 }
```

17.2.2. Paso de Computador desarrollador a Raspberry para ejecutar

Una vez instalado el controlador dotnet en el hardware se puede ejecutar una aplicación dependiente del marco o ejecutar un comando desde la línea de comandos. Para poder ejecutar este programa se publica el archivo del proyecto (Ver figura 17.3) en una carpeta local para poder pasar los archivos a una carpeta dentro de la Raspberry.

Figura 17.3. Opción para publicar el proyecto desde visual studio



Fuente: Propia

Se publica en una carpeta local, donde se generan diferentes archivos del programa:

- .exe
- .dll
- .deps.json
- .pbd
- .runtimeconfig.json

Para ejecutar el programa realizado se pasa el archivo .dll a una carpeta en el dispositivo (Raspberry pi 3) donde contiene la base de datos procesada para la lectura de signos vitales (llamada "test.txt"). Posteriormente, se le concede permiso de ejecutar al archivo .dll, con el fin de que desde el terminal se envíe el comando "*dotnet NombrePrograma*.dll*" para ejecutar el simulador.

Capítulo 18

Resultados

Finalmente, juntando todas las partes del proyecto se obtiene el sistema de información el cual consta de una aplicación web (<https://information-system-project.herokuapp.com/>), una base de datos alojada en la nube, un simulador de señales incorporado en una tarjeta de desarrollo como lo es la Raspberry Pi 3 y la generación de alertas para las anomalías de los signos vitales.

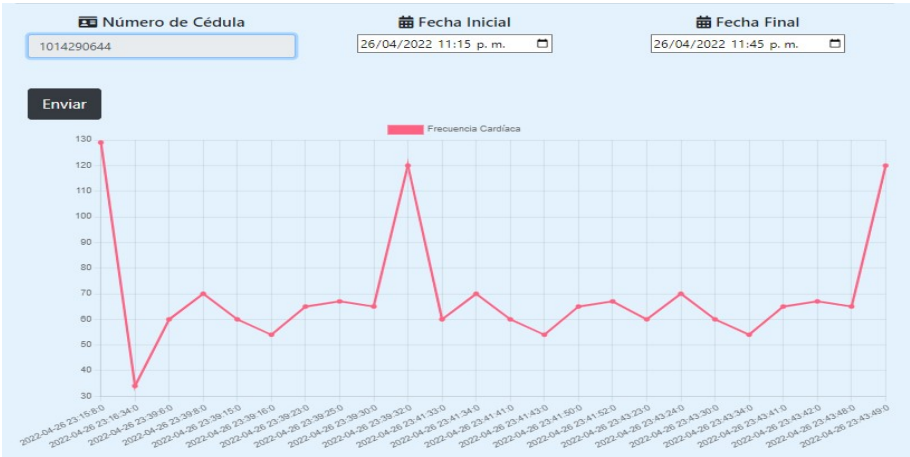
Partiendo de esto, se va a mostrar uno a uno el sistema de información cumpliendo con lo estipulado en el planteamiento.

18.1. Visualizar

La parte visual del proyecto se valida entrando a la aplicación web, ya sea como Usuario o como Analista, y seleccionando un rango de tiempo en el historial de datos para generar las gráficas en donde se muestran los datos almacenados de los signos vitales: Frecuencia Cardiaca (ver figura

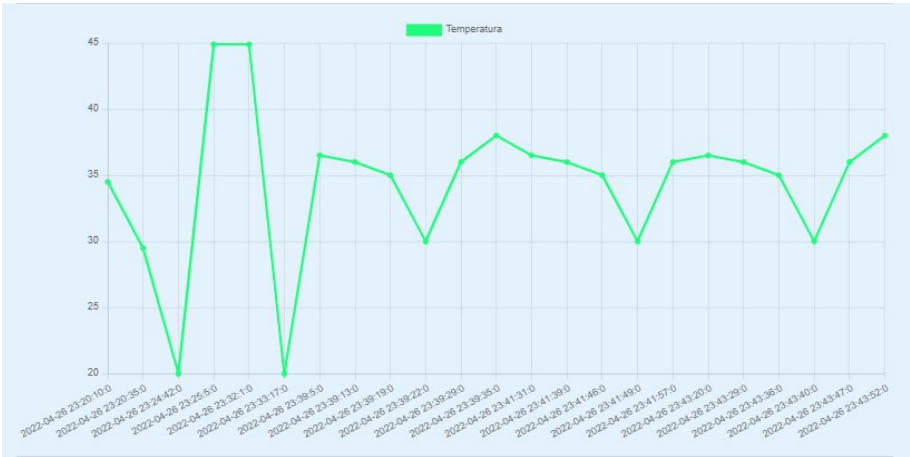
18.1), Temperatura (ver figura 18.2) y Oximetría (ver figura 18.3)

Figura 18.1. Gráfica Frecuencia Cardíaca



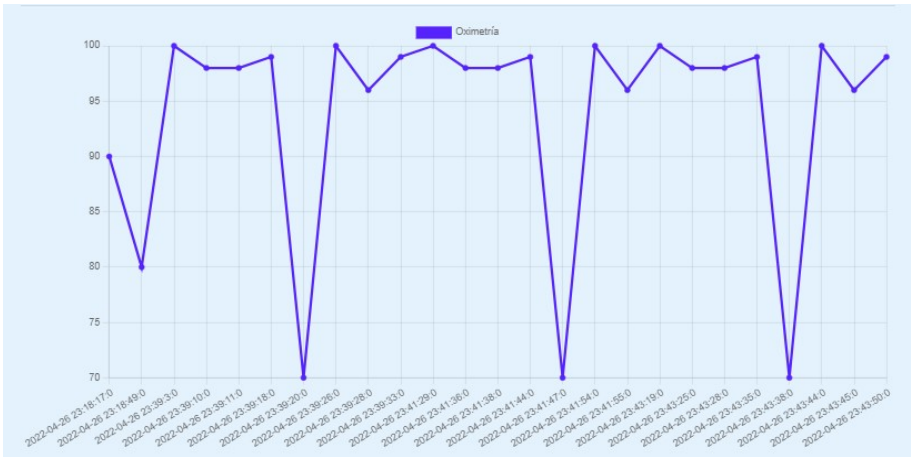
Fuente: Propia

Figura 18.2. Gráfica Temperatura



Fuente: Propia

Figura 18.3. Gráfica Oxigenación en la Sangre



Fuente: Propia

Además de ver las gráficas de los signos vitales, ya sea en historial o en vivo, otra parte visual del proyecto a validar es la tabla del historial de alertas, la cual tiene el signo vital relacionado, la anomalía presentada y la hora y fecha en la que sucedió como se puede ver en la figura 18.4.

Figura 18.4. Tabla de anomalías presentadas

Número de Cédula

1014290644

Enviar

Signo Vital	Anomalia Registrada	Fecha de Registro
Temperatura	Hipertermia	Tue Apr 26 2022 23:32:01 GMT-0500 (Colombia Standard Time)
Temperatura	Hipotermia grave	Tue Apr 26 2022 23:33:17 GMT-0500 (Colombia Standard Time)
Frecuencia Cardíaca	Latidos cardíacos lentos (bradicardia)	Tue Apr 26 2022 23:43:34 GMT-0500 (Colombia Standard Time)
Temperatura	Hipotermia leve	Tue Apr 26 2022 23:43:36 GMT-0500 (Colombia Standard Time)
Oximetría	Grave desaturación	Tue Apr 26 2022 23:43:38 GMT-0500 (Colombia Standard Time)
Temperatura	Hipotermia	Tue Apr 26 2022 23:43:40 GMT-0500 (Colombia Standard Time)
Frecuencia Cardíaca	Latidos del corazón rápidos (taquicardia)	Tue Apr 26 2022 23:43:49 GMT-0500 (Colombia Standard Time)
Temperatura	Hipertermia	Tue Apr 26 2022 23:43:52 GMT-0500 (Colombia Standard Time)
Temperatura	Hipotermia leve	Wed Apr 27 2022 01:00:03 GMT-0500 (Colombia Standard Time)
Oximetría	Grave desaturación	Wed Apr 27 2022 01:00:05 GMT-0500 (Colombia Standard Time)
Frecuencia Cardíaca	Latidos del corazón rápidos (taquicardia)	Wed Apr 27 2022 01:00:18 GMT-0500 (Colombia Standard Time)
Temperatura	Hipertermia	Wed Apr 27 2022 01:00:22 GMT-0500 (Colombia Standard Time)
Temperatura	Hipotermia leve	Wed Apr 27 2022 01:01:51 GMT-0500 (Colombia Standard Time)
Oximetría	Grave desaturación	Wed Apr 27 2022 01:01:53 GMT-0500 (Colombia Standard Time)

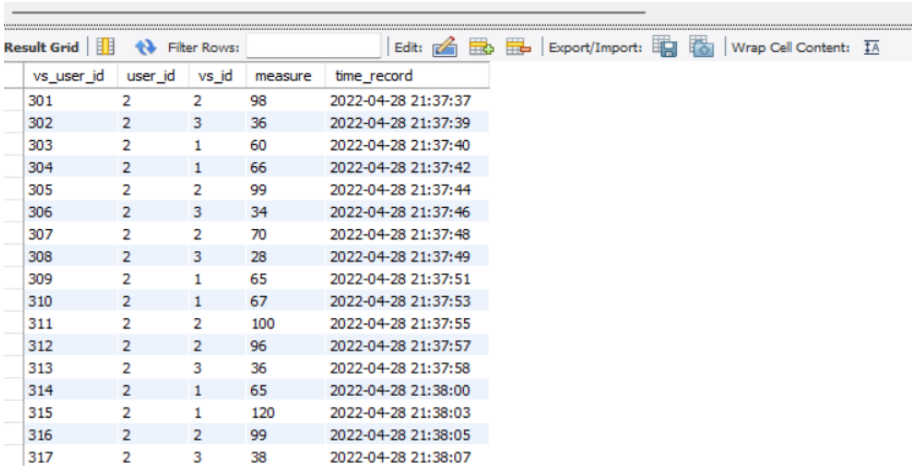
Fuente: Propia

18.2. Almacenar

Se valida el almacenamiento de la información generada por el simulador desde el MySQL Workbench, donde se puede ver el ingreso de información de signos vitales, usuarios e historial de alertas. (Ver figuras 18.5 , 18.6 y 18.7)

Figura 18.5. Tabla de Signos vitales de Usuarios

```
1 • SELECT * FROM heroku_a72350352312ba8.vital_signs_users;
```



vs_user_id	user_id	vs_id	measure	time_record
301	2	2	98	2022-04-28 21:37:37
302	2	3	36	2022-04-28 21:37:39
303	2	1	60	2022-04-28 21:37:40
304	2	1	66	2022-04-28 21:37:42
305	2	2	99	2022-04-28 21:37:44
306	2	3	34	2022-04-28 21:37:46
307	2	2	70	2022-04-28 21:37:48
308	2	3	28	2022-04-28 21:37:49
309	2	1	65	2022-04-28 21:37:51
310	2	1	67	2022-04-28 21:37:53
311	2	2	100	2022-04-28 21:37:55
312	2	2	96	2022-04-28 21:37:57
313	2	3	36	2022-04-28 21:37:58
314	2	1	65	2022-04-28 21:38:00
315	2	1	120	2022-04-28 21:38:03
316	2	2	99	2022-04-28 21:38:05
317	2	3	38	2022-04-28 21:38:07

Fuente: Propia

Figura 18.6. Tabla de Usuarios

1 SELECT * FROM heroku_a72350352312ba8.users;

Result Grid										
Filter Rows:										
Edit: Export/Import: Wrap Cell Content:										
user_id	email	password	name	lastname	identification	birthday	phone	role_id	gender	
1	lagossergio0@gmail.com	\$2a\$10\$/J11.2old554oMfok/Bf.kcQzSMYf9d8...	Sergio	Lagos Fuquen	1014304188	1999-03-13	3013655421	2	masculino	
2	mateosparra@usantotomas.edu.co	\$2a\$10\$/7k9VLUVRZdFhZ5OOqauJMcR6FhtP...	Mateo	Parra Posse	1014290644	1997-01-01	3007636582	1	masculino	
3	spidy@gmail.com	\$2a\$10\$/5Pj7dXgR5w2Ksbr7JA9eWlocJIEIfrr...	Peter	Parker	1012569833	2001-08-01	3102874569	1	masculino	
4	ironman@gmail.com	\$2a\$10\$/M3rURqgkUQu7yrEUAU9ccJe6i.vU9JN6...	Tony	Stark	1023569877	1995-05-03	3176645599	2	masculino	
5	dariosegura@usantotomas.edu.co	\$2a\$10\$/pV1yZdHGySTqUWScrXh.qMmqEsv7...	Dario	Segura	2118540	1978-11-01	3165241941	2	masculino	

Fuente: Propia

Figura 18.7. Tabla de historial de alertas

1 SELECT * FROM heroku_a72350352312ba8.alert_history;

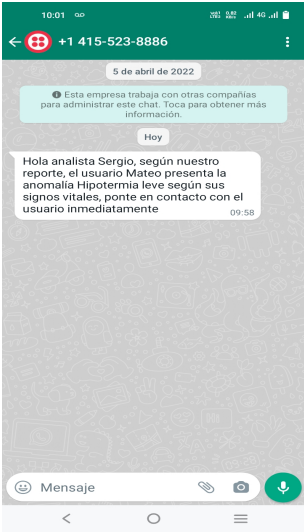
Result Grid										
Filter Rows:										
Edit: Export/Import: Wrap Cell Content:										
alert_hist_id	time_record	user_id	anomaly_id							
1	2022-04-26 23:32:01	2	8							
2	2022-04-26 23:33:17	2	7							
3	2022-04-26 23:43:34	2	2							
4	2022-04-26 23:43:36	2	5							
5	2022-04-26 23:43:38	2	4							
6	2022-04-26 23:43:40	2	6							
7	2022-04-26 23:43:49	2	1							
8	2022-04-26 23:43:52	2	8							
9	2022-04-27 01:00:03	2	5							
10	2022-04-27 01:00:05	2	4							

Fuente: Propia

18.3. Generar alertas

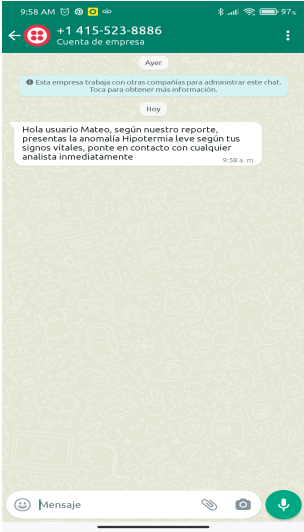
La validación de la generación de alertas se realiza mediante el mensaje generado en el programa y que el bot de Twilio envía a los Analistas y a los Usuarios vía WhatsApp como se puede ver en las figuras 18.8 y 18.9 respectivamente.

Figura 18.8. Mensaje Analista



Fuente: Propia

Figura 18.9. Mensaje Usuario



Fuente: Propia

Conclusiones

- Se desarrolló un sistema de información basado en el lenguaje JavaScript, que permite la creación de usuarios con roles de analistas o pacientes, los cuales tienen la posibilidad de visualizar a través de gráficas, sus signos vitales propios (usuario: paciente) o de los pacientes actualmente registrados (usuario: analista) de manera versátil en cualquier dispositivo con acceso a internet, sin limitarse a un host local que tenga la información centralizada, evitando así gastos de hardware. También, el sistema de información muestra un historial de anomalías detectadas en el momento del registro de los datos recibidos, para llevar un control de tiempo de irregularidades presentadas en sus signos.
- Se configuró por medio de una herramienta innovadora, una alerta personalizada para pacientes y analistas; que informa las anomalías puntuales evidenciadas en el momento de registro de los datos recibidos por el sistema a través de la aplicación WhatsApp.
- Se construyó un simulador de información empleando lenguaje C# para enviar los datos adaptados desde una base de datos de signos vitales publica (Physionet); es ejecutado en una tarjeta Raspberry Pi 3 Modelo B, para imitar el comportamiento de la información de un paciente para alimentar el sistema.

- Se generaron dos (2) tablas maestras en la base de datos: "vital_signs" y "anomalies" las cuales buscan permitir al sistema de información ser escalable, para seguir integrando más bioseñales, anomalías y diferentes tipos de usuarios al momento de mejorar el sistema, es decir, cuando aumenten sus funcionalidades.
- Al desplegar la aplicación web en la nube y simulando el ingreso de datos por un método asíncrono de la ruta estipulada, se observa que no hay incidencias en recibir una o más conexiones al momento de recibir datos, abriendo así la posibilidad de usar un captador de varias señales o de un signo vital en específico, por usuario conectado al sistema; dando versatilidad en el diseño del ingreso de datos.
- Se observaron limitantes frente a Heroku, ya que al ser una plataforma de servicios en la nube, esta genera restricciones cuando se usa una cuenta gratuita, como por ejemplo, que tiene un número máximo de solicitudes por hora hacia la base datos de MySQL. Cabe recalcar que si se desea continuar un proyecto similar a largo plazo, se tiene que realizar el pago oportuno del servicio.

Bibliografía

- [1] O. M. de la Salud, “La esperanza de vida ha aumentado en 5 años desde el año 2000, pero persisten las desigualdades sanitarias,” <https://www.who.int/es/news/item/19-05-2016-life-expectancy>, 19 de Mayo de 2016.
- [2] E. de investigación Enterarse, “¿cada vez vivimos más? la evolución de la esperanza de vida en el mundo,” <https://www.enterarse.com/202102150001-cada-vez-vivimos-mas>, Febrero 2021.
- [3] B. I. de Desarrollo, “Detección, prevención, respuesta y recuperación con tecnología digital,” <https://publications.iadb.org/publications/spanish/document/Deteccion-prevencion-respuesta-y-recuperacion-con-tecnologia-digital>, 2020.
- [4] Blogoeconomía, “La silla vacía,” Octubre 2017, <https://lasillavacia.com/blogs/blogoeconomia/el-sistema-de-salud-colombiano-bueno-o-malo-62792>.
- [5] Semana, “¿qué pasa con la calidad del sistema de salud en colombia?” Octubre 2018, <https://www.semana.com/pais/articulo/cual-es-el-problema-del-sistema-de-salud-colombiano/263731/>.
- [6] Dinero, Enero 2018, <https://www.dinero.com/pais/articulo/cual-es-el-problema-del-sistema-de-salud-colombiano/263731>.

- [7] F. C. Suárez Cadena, "Manteniendo la relación médico-paciente a pesar de las dificultades del sistema de salud Colombiano," *Medicas UIS*, vol. 29, pp. 7 – 9, 08 2016. [Online]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-03192016000200001&nrm=iso
- [8] M. Camilo, B. Eric, B. William, and E. Germán, "Ministerio de salud," <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/CA/Ranking-satisfaccion-eps-2018.pdf>, Marzo 2018.
- [9] S.N., "Android apis," , 04 2021.
- [10] S. M. Fredy and R. A. Rafael, *Diseño e implementación de un sistema para el almacenamiento de historial clínico en tarjetas inteligentes*. Universidad de Los Andes, 2005.
- [11] M. Branzila and V. David, "Wireless intelligent systems for biosignals monitoring," *Proceedings of the 3rd International Conference on E-Health and Bioengineering - EHB 2011*, pp. 1–4, 2011.
- [12] T. E. Ismael, *Diseño y elaboración de un Prototipo de monitor de signos vitales aplicando métodos no invasivos con comunicación de datos a dispositivos móviles*. Universidad Politécnica Salesiana, 2015. [Online]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/7982/1/UPS-CT004847.pdf>
- [13] R. Zaic, *Monitor de Signos Vitales Portátil*. Universidad Autónoma de Guerrero, 2016. [Online]. Available: http://ri.uagro.mx/bitstream/handle/uagro/295/OK14638639_maestria_1.pdf?sequence=1&isAllowed=y
- [14] S. Óscar and R. E. John, *Monitoreo remoto de signos corporales y transmisión de datos y alertas a una aplicación instalada en un smartphone*. Universidad Distrital Francisco José de Caldas, 2018. [Online]. Available: <http://repository.udistrital.edu.co/bitstream/11349/13383/2/SarmientoGómezOscar2018.pdf>
- [15] Koogeeek, "Koogeeek ksbp2," 2018, <https://www.koogeeek.com/p-ksbp2-1.html>.

- [16] P. Medline, "Medline plus," <https://medlineplus.gov/spanish/ency/article/002341.htm>, 2015.
- [17] R. Susana and R. Eva, *Fundamentos de Enfermería. El Manual Moderno*, 2004.
- [18] N. Soengas, "Alteraciones en el control de la temperatura corporal en los adultos mayores," *Publicaciones Latinoamericanas*, vol. 12, pp. 34 – 36, 06 2018. [Online]. Available: http://adm.meducatium.com.ar/contenido/articulos/16000340036_1125/pdf/16000340036.pdf
- [19] M. Marina, M. Javier, B. Antonio, and A. Felipe, <http://www.medynet.com/usuarios/jraguilar/>, 2017.
- [20] R. Cabrera Montesinos, M. Castro Arias, J. L. Tápanes Morejón, and G. Durán Arrieta, "Método físico para la regulación de la temperatura corporal ," *Revista Cubana de Enfermería*, vol. 13, pp. 80 – 85, 12 1997. [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-03191997000200004&nrm=iso
- [21] G. D. Medical, "Gräger," <https://www.draeger.com/Library/Content/>, 2011.
- [22] S. Daniel, "Temperature monitoring and perioperative thermoregulation." *PubMed*, 2008.
- [23] N. A. Cotrina Araujo, "Análisis predictivo de series temporales de temperatura corporal," Master's thesis, Pontificia Universidad Católica del Perú, Perú, 11 2020, <http://hdl.handle.net/20.500.12404/17511>.
- [24] L. Veloza, C. Jiménez, D. Quiñones, F. Polanía, L. C. Pachón-Valero, and C. Y. Rodríguez-Triviño, "Variabilidad de la frecuencia cardiaca como factor predictor de las enfermedades cardiovasculares," *Revista Colombiana de Cardiología*, vol. 26, no. 4, pp. 205–210, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0120563319300683>
- [25] M. Markus, "Medicla news today," Enero 2019, <https://www.medicalnewstoday.com/articles/291182.php>.

- [26] E. Generación, "10 puntos de palpación de los pulsos arteriales," 04 2017, <https://www.elsevier.com/es-es/connect/medicina/10-puntos-de-palpacion-de-los-pulsos-arteriales>.
- [27] G. Jaime, F. Jorge, and Álvarez Diego, "Análisis espectral de la variabilidad de la frecuencia cardíaca." *Iatreia*, vol. Retrieved from <https://revistas.udea.edu.co/index.php/iatreia/article/view/3682>, 1999.
- [28] I. Cabrera Rojo, A. Cabrera Santos, and G. Gallardo Montes de Oca, "Variabilidad de la frecuencia cardíaca en el joven normal ," *Revista Cubana de Investigaciones Biomédicas* , vol. 16, pp. 98 – 103, 12 1997. [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-03001997000200004&nrm=iso
- [29] E.-Z. N. L.-U. O. T.-D. J. D. I. R.-I. J. M.-S. D. De la Torre-Paz JE, Mo-Ye G, "Electrocardiograma vs electrocardiograma celular en el servicio de Urgencias ," *Medicina Interna de México* , vol. 37, pp. 173 – 181, 04 2021. [Online]. Available: <https://www.medigraphic.com/pdfs/medintmex/mim-2021/mim212c.pdf>
- [30] P. de Mayo Clinic, "Arritmia cardíaca," , 2021.
- [31] V. G. Juliana, V. A. A. Oscar, and V. G. Valentina, "Semiología de los signos vitales: Una mirada novedosa a un problema vigente," *Archivos de Medicina (Col)*, vol. 12, pp. 221 – 240, 2012.
- [32] P. A. Oscar, "Hemoglobina: una molécula modelo para el investigador," in *Colombia Médica*. Red Colombia Médica, 2006, pp. 215 – 225.
- [33] O. M. M., B. Hernando, A. R. Jorge, and D. L. P. S. I., "Índice de perfusión periférica en la uci neonatal: una respuesta a la monitorización no invasiva del recién nacido crítico," *Perinatología y Reproducción Humana*, vol. 31, pp. 85 – 90, 2017.
- [34] S.N., "Introducción a .net framework," , 04 2022.
- [35] S. Rod, *Begining Database Design Solutions*. Wrox, 2008.
- [36] S. Abraham, K. H. F., and S. S., *FUNDAMENTOS DE BASES DE DATOS*. Concepción Fernández Madrid, 2010.

- [37] E. pensante, "Ventajas y desventajas de las bases de datos," , 2015.
- [38] P. G. M. José, *Diseño de Bases de Datos Relacionales*. Ediciones Paraninfo, S.A., 2014.
- [39] C. Y. Angel, *Diseño y Programación de Bases de Datos*. Visión Libros, 2007.
- [40] H. D. Center, "Stacks," , 2022.
- [41] G. Gonzales, "Los mejores servicios que ofrecen alojamientos gratis para tus webs, aplicaciones o experimentos," , 03 2022.
- [42] C. ES, "¿qué es twilio y cómo puede funcionar para su negocio?" , 08 2018.
- [43] T. Dimes, *JavaScript Una Guía de Aprendizaje para el Lenguaje de Programación JavaScript*. Babelcube, Inc, 2014.
- [44] S.N., "Raspberry pi 3 model b," , 02 2016.
- [45] —, "Raspberry pi 3," , 2016.
- [46] W. R. Ruiz Sánchez, "Redes de sensores inalámbricos enfocadas a la medicina con énfasis en control de los signos vitales en pacientes adultos mayores," masterThesis, Pontificia Universidad Católica del Ecuador, Ecuador, 09 2016, <http://repositorio.puce.edu.ec/handle/22000/11311>.
- [47] M. L. Luna-Gonzalez, S. M. Becerra-Bayona, N. Serrano-Diaz, and R. A. Lobo-Quintero, "Implementación de tecnologías libres y sensores remotos para un biobanco: el desafío de producir a bajo costo," *Información tecnológica*, vol. 31, pp. 149 – 162, 04 2020. [Online]. Available: http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642020000200149&nrm=iso
- [48] B. J. Aguirre Ruiz, "Evaluación de las interfaces de un sistema de monitoreo remoto de pacientes con diabetes utilizando una guía de usabilidad," masterThesis, Universidad Autónoma de Ciudad Juárez, México, 09 2017, <http://hdl.handle.net/20.500.11961/3080>.

- [49] S. L. Hsieh, F. Lai, P. H. Cheng, J. L. Chen, H. H. Lee, W. N. Tsai, Y. C. Weng, S. H. Hsieh, K. P. Hsu, L. F. Ko, T. H. Yang, C. H. Chen, and C. H. Chen, "An integrated healthcare enterprise information portal and healthcare information system framework," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006, pp. 4731–4734.
- [50] M. H. Selamat, M. S. Othman, N. H. M. Shamsuddin, N. I. M. Zukepli, and A. F. Hassan, "A review on open source architecture in geographical information systems," in *2012 International Conference on Computer Information Science (ICCIS)*, vol. 2, 2012, pp. 962–966.
- [51] J. M. M. Ospina, M. F. Sánchez, N. B. Castilla, and L. F. Wanumen, "Sistema de alertas de signos vitales y de ubicación a través de dispositivos móviles."
- [52] A. d. J. Guzmán Clemente and O. A. Bernardo Mazariegos, "Sistema web para el monitoreo de signos vitales y control del historial clínico de pacientes hospitalizados en servicios de medicina interna," Tech. Rep., 2014.
- [53] J. E. González-Barajas, E. Francisco-Forero, and I. Marino-Martinez, "Tratamiento de rizados en la estimación de la densidad espectral de potencia de la señal de ritmo cardiaco," *Revista UIS Ingenierías*, vol. 12, no. 2, p. 17–27, nov. 2013. [Online]. Available: <https://revistas.uis.edu.co/index.php/revistausingenierias/article/view/17-27>
- [54] U. Fundación, "Javascript: el lenguaje de programación fundamental para la creación de webs," , Noviembre 2021.
- [55] Desarrolloweb.com, "Manual de handlebars," .
- [56] G. Author, "Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo," , 04 2020.
- [57] Pulzo, "Cinco tendencias que su negocio debería tener en cuenta al interactuar con clientes," , 04 2022.
- [58] M. Build, "¿qué es git?" , 2022.

- [59] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
- [60] J. Birjandtalab, D. Cogan, M. B. Pouyan, and M. Nourani, "A non-ecg biosignals dataset for assessment and visualization of neurological status," in *2016 IEEE International Workshop on Signal Processing Systems (SiPS)*, Oct 2016, pp. 110–114.
- [61] D. C. M. B. P. Birjandtalab, Javad and M. Nourani, "A non-ecg biosignals dataset for assessment and visualization of neurological status," pp. 110–114, 2016.
- [62] D. Mongin, C. Chabert, D. S. Courvoisier, J. García-Romero, and J. R. Alvero-Cruz, "Heart rate recovery to assess fitness: comparison of different calculation methods in a large cross-sectional study," *Research in Sports Medicine*, vol. 0, no. 0, pp. 1–14, 2021, pMID: 34308736. [Online]. Available: <https://doi.org/10.1080/15438627.2021.1954513>
- [63] J. R. A. C. Denis Mongin, Jeronimo García Romero, "Treadmill maximal exercise tests from the exercise physiology and human performance lab of the university of malaga," 2021.
- [64] A. S. Bhogal and A. R. Mani, "Pattern analysis of oxygen saturation variability in healthy individuals: Entropy of pulse oximetry signals carries information about mean oxygen saturation," *Frontiers in Physiology*, vol. 8, 2017. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fphys.2017.00555>
- [65] A. L. G. L. H. J. I.-P. C. M.-R. . . S. H. E. Goldberger, A., "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals." pp. 215–220, 2000.
- [66] M. Build, "Implementación de aplicaciones .net en raspberry pi," , 04 2022.

Anexos

Anexo A: Script para creación de Base de Datos SQL del sistema

Código 18.1. Script creación Base de Datos.

```
1 --Fuente: Propia.
2 use heroku_a72350352312ba8;
3
4 SET @@auto_increment_increment=1;
5 CREATE TABLE roles (
6     role_id BIGINT NOT NULL AUTO_INCREMENT,
7     role VARCHAR(20) NOT NULL,
8     PRIMARY KEY (role_id)
9 );
10
11 SET @@auto_increment_increment=1;
12 CREATE TABLE users (
13     user_id BIGINT NOT NULL AUTO_INCREMENT,
14     email VARCHAR(100) NOT NULL,
15     password VARCHAR(100) NOT NULL,
16     name VARCHAR(100) NOT NULL,
17     lastname VARCHAR(100) NOT NULL,
18     identification BIGINT NOT NULL,
19     birthday DATE NOT NULL,
20     phone BIGINT NOT NULL,
21     role_id BIGINT NOT NULL,
22     gender VARCHAR(20) NOT NULL,
23     PRIMARY KEY (user_id),
```

```

24  CONSTRAINT fk_role_users FOREIGN KEY(role_id) REFERENCES roles(
      role_id)
25 );
26
27 SET @@auto_increment_increment=1;
28 CREATE TABLE vital_signs (
29     vs_id BIGINT NOT NULL AUTO_INCREMENT,
30     vital_sign VARCHAR(20) NOT NULL,
31     PRIMARY KEY (vs_id)
32 );
33
34 SET @@auto_increment_increment=1;
35 CREATE TABLE anomalies (
36     anomaly_id int(100) NOT NULL AUTO_INCREMENT,
37     vs_id BIGINT NOT NULL,
38     max_rate float NOT NULL,
39     min_rate float NOT NULL,
40     gender VARCHAR(20) NOT NULL,
41     age int(100) NOT NULL,
42     description VARCHAR(100) NOT NULL,
43     PRIMARY KEY (anomaly_id),
44     CONSTRAINT fk_vs_anom FOREIGN KEY(vs_id) REFERENCES vital_signs
      (vs_id)
45 );
46
47 SET @@auto_increment_increment=1;
48 CREATE TABLE alert_history (
49     alert_hist_id int NOT NULL AUTO_INCREMENT,
50     time_record datetime NOT NULL,
51     user_id BIGINT NOT NULL,
52     anomaly_id int(100) NOT NULL,
53     PRIMARY KEY (alert_hist_id),
54     CONSTRAINT fk_user_hist FOREIGN KEY(user_id) REFERENCES users(
      user_id),
55     CONSTRAINT fk_anom_hist FOREIGN KEY(anomaly_id) REFERENCES
      anomalies(anomaly_id)
56 );
57
58 SET @@auto_increment_increment=1;
59 CREATE TABLE vital_signs_users (
60     vs_user_id BIGINT NOT NULL AUTO_INCREMENT,
61     user_id BIGINT NOT NULL,
62     vs_id BIGINT NOT NULL,
63     measure float NOT NULL,
64     time_record datetime NOT NULL,

```

```

65 PRIMARY KEY (vs_user_id),
66 CONSTRAINT fk_user_vsu FOREIGN KEY(user_id) REFERENCES users(
  user_id),
67 CONSTRAINT fk_vs_vsu FOREIGN KEY(vs_id) REFERENCES vital_signs
  (vs_id)
68 );
69
70 -- Roles:
71 insert into roles(role) values ('user');
72 insert into roles(role) values ('analist');
73 -- Signos vitales:
74 insert into vital\_signs(vital_sign) values ('Frecuencia
  Card aca');
75 insert into vital\_signs(vital_sign) values ('Oximetr a') ;
76 insert into vital\_signs(vital_sign) values ('Temperatura') ;
77 }

```

Anexo B: Generación de alertas mediante el código de la aplicación

Código 18.2. Alertas mediante el código de la aplicación.

```

78 \\Fuente: Propia.
79 const client = require('twilio')("
  AC24d6cbd97e4536d4b5b1e797752ff69c", "848
  e9343dde7d345f47954f6cc96b78b");
80 anomaliesQuery = await pool.query('SELECT anomaly_id,
  description FROM anomalies WHERE vs_id = ? AND max_rate > ?
  AND min_rate < ?', [modelSignalDb.vs_id, modelSignalDb.measure,
  modelSignalDb.measure] );
81
82 if (anomaliesQuery[0] !== undefined){
83   let analist = await pool.query('SELECT phone, name FROM users
  WHERE role_id="2"');
84   let user = await pool.query('SELECT phone, name FROM users
  WHERE user_id = ?', [modelSignalDb.user_id]);
85
86   analist.forEach(element => {
87     console.log('celular analistas', element.phone)
88     client.messages.create({
89       from: 'whatsapp:+14155238886',

```

```

90         body: 'Hola analista ' + element.name + ', seg n
nuestro reporte, el usuario ' + user[0].name + ' presenta la
anomal a '+ anomaliesQuery[0].description + ' seg n sus
signos vitales, ponte en contacto con el usuario
inmediatamente',
91         to: 'whatsapp:+57' + element.phone,
92     })).then(message => console.log(message.sid));
93 });
94 console.log('celular usuario ', user[0].phone);
95 client.messages.create({
96     from: 'whatsapp:+14155238886',
97     body: 'Hola usuario ' + user[0].name + ', seg n nuestro
reporte, presentas la anomal a '+ anomaliesQuery[0].
description + ' seg n tus signos vitales, ponte en contacto
con cualquier analista inmediatamente',
98     to: 'whatsapp:+57' + user[0].phone,
99 }).then(message => console.log(message.sid));

```

Anexo C: Generación de alertas mediante el código de la aplicación

Código 18.3. Programa C# Simulador de señales.

```

100
101 using System;
102 using System.IO;
103
104 [Serializable]
105 public class VSNotFoundException : Exception
106 {
107     public VSNotFoundException() { }
108     public string newVS { get; }
109
110     public VSNotFoundException(string message)
111         : base(message) { }
112
113 }
114
115 namespace MyApp
116 {
117     internal class Program

```

```
118     {
119         static void Main(string[] args)
120         {
121             String line, id_user, id_vs, measure;
122
123
124             try
125             {
126                 //Se pasa la ruta del archivo y el nombre del
archivo al constructor de StreamReader
127                 StreamReader sr = new StreamReader("../test.txt")
;
128                 //Le la primera linea de texto.
129                 line = sr.ReadLine();
130
131                 //Continua leyendo hasta llegar al final del
archivo
132                 while (line != null)
133                 {
134                     //se toma el ID del usuario
135                     id_user = line.Substring(2, line.IndexOf(";"),
0) - 2);
136                     //se toma el ID del signo vital, dependiendo
la letra
137                     switch (line.Substring(line.IndexOf(";"), 0) +
1, 1))
138                     {
139                         case "F": //Frecuencia Cardiaca 1
140                             id_vs = "1";
141                             break;
142                         case "O": //Oximetria 2
143                             id_vs = "2";
144                             break;
145                         case "T": //Temperatura 3
146                             id_vs = "3";
147                             break;
148                         default:
149                             throw new VSNotFoundException("No
existe signo vital");
150                             break;
151                     }
152                     //se toma el valor dle singo vital
153                     measure = line.Substring(line.IndexOf(";"), 0)
+ 2, line.Length - (line.IndexOf(";"), 0) + 2));
154
```

```
155         //se crea la clase para la solicitud http
156         HttpClient cliente = new HttpClient();
157
158         //Se envia la colicitud HTTP con los nuevos
valores
159         var resp = cliente.GetAsync("https://
information-system-project.herokuapp.com/information-data?
user_id=" +
160             id_user +
161             "&vs_id=" +
162             id_vs +
163             "&measure=" +
164             measure).Result;
165
166         //Le la siguiente linea del archivo txt
167         line = sr.ReadLine();
168         Thread.Sleep(1000);
169     }
170     //Cierra el stream reader, una vez acabado el
documento de lectura
171     sr.Close();
172     Console.ReadLine();
173 }
174 catch (Exception e)
175 {
176     Console.ForegroundColor = ConsoleColor.Red;
177     Console.WriteLine("Exception: " + e.Message);
178 }
179 finally
180 {
181     Console.WriteLine("Executing finally block.");
182 }
183
184 }
185 }
186 }
```