



**Instituto Politécnico Nacional
Escuela Superior de Cómputo**



**SISTEMA DE MONITOREO DE PULSO, TEMPERATURA Y
SATURACIÓN DE OXÍGENO PARA SEGUIMIENTO DE
PACIENTES EN CENTROS DE SALUD “DOC-CHECKER”**

Trabajo Terminal

No. 2021-A047

Alumnos:

Conde Francisco José Ángel

Domínguez Reyes Jesús Alejandro

Nambo Velázquez Carlos

**Velasco Martínez Alan Alexis*

Directoras:

Ocampo Botello Fabiola

Fabiola Ocampo Botello

Rocha Bernabé María del Rosario *V. Bo. Rosario Rocha B.*

**email: avelascom1701@alumno.ipn.mx*

Capítulo I: Estudio preliminar	4
1.1 Introducción	4
1.2 Objetivo general	5
1.2.1 Objetivos específicos	5
1.3 Justificación	5
1.4 Productos o Resultados esperados	6
1.5 Metodología	6
1.6 Estado del Arte	7
Capítulo II: Marco teórico	11
2.1 Signos vitales y su importancia	11
2.1.2 Temperatura corporal	11
2.1.3 Frecuencia cardiaca	12
2.1.4 Oximetría	12
2.1.5 Monitoreo de los signos vitales / sensores	13
2.2 Servidores	13
2.2.1 Sockets	13
2.3 Internet de las Cosas (IoT, por sus siglas en inglés)	14
2.4 Sistemas Embebidos	14
2.5 Aplicaciones Híbridas	14
2.6 Bases de datos	14
2.6.1 Tipos de bases de datos	14
2.7 Minería de Datos	15
Capítulo III: Análisis de Riesgos	16
3.1 Identificación de Riesgos	16
3.1.1 Jerarquización	19
3.1.2 Gestión de riesgos	20
3.2 Métricas y Estimación	22
3.2.1 Métrica Puntos de Función	22
3.2.2 COCOMO II	24
3.2.3 Costo total del Software	25
Capítulo IV: Diseño de la solución propuesta	26
4.1 Análisis de requerimientos del sistema	26
4.1.1 Requerimientos funcionales	26
4.1.2 Requerimientos no funcionales	26
4.2 Análisis de requerimientos Software	27
4.2.1 Análisis de requerimientos de la aplicación híbrida	27
4.2.1.1 Selección del lenguaje de programación	27

4.2.1.2 Selección del Framework	28
4.2.1.3 Selección del entorno de desarrollo	30
4.2.2 Análisis de requerimientos del servidor	32
4.2.2.1 Selección del lenguaje de programación	32
4.2.2.2 Selección del Sistema Gestor de Bases de Datos	33
4.2.2.3 Selección del entorno de desarrollo	35
4.3 Análisis de requerimientos Hardware (módulo de monitoreo)	36
4.3.1 Selección de sensor de temperatura	36
4.3.2 Selección de sensor para oximetría	38
4.3.3 Selección de sensor de frecuencia cardiaca	38
4.3.4 Selección de la placa de desarrollo	39
4.3.5 Selección del lenguaje de programación para la placa	41
4.3.6 Selección del entorno de desarrollo	42
4.4 Diagrama de Casos de Uso	44
4.5 Diagramas de Secuencia	44
4.5.1 Diagrama de Secuencia “Registro de Médico/Enfermero”	44
4.5.2 Diagrama de Secuencia “Registro de Paciente”	45
4.5.3 Diagrama de Secuencia “Monitoreo de Signos Vitales”	46
4.5.4 Diagrama de Secuencia “Consultar reporte “Minería de Datos””	47
4.6 Diagramas de Actividades	48
4.6.1 Diagrama de Actividad “Registro de Médico/Enfermero”	48
4.6.2 Diagrama de Actividad “Registro del paciente”	49
4.6.3 Diagrama de Actividad “Monitoreo de Signos Vitales”	50
4.6.4 Diagrama de Actividad “Consultar reporte “Minería de Datos””	51
4.7 Diagramas de flujo de datos	52
4.7.1 Diagrama de contexto	52
4.7.2 Diagrama de flujo de datos lógico	53
4.8 Diseño de la base de datos	54
4.8.1 Diagramas base de datos	54
4.8.2 Diagrama entidad-relación	54
4.8.3 Modelo Relacional	55
4.9 Comunicación de los componentes	56
4.10 Mockups de la Aplicación Híbrida	57
4.10.1 Inicio de Sesión	57
4.10.2 Registro	59
4.10.3 Registro Pacientes	61
4.10.4 Home (Lista de pacientes)	66
4.10.5 Signos Vitales en tiempo real	67

4.10.6 Informe de paciente	69
4.10.7 Configuración	71
4.10.8 Consulta de Reporte de Minería de Datos	73
4.10.9 Búsqueda de pacientes	76
4.10.10 Pestaña administrador	77
4.10.11 Solicitudes de Médicos	78
4.10.12 Ingreso de placas	79
4.10.13 Búsqueda de pacientes en administrador	80
4.10.14 Búsqueda de Médicos en administrador	80
4.10.15 Búsqueda de placas en administrador	81
4.11 Diagrama de flujo del algoritmo a programar en la placa de desarrollo	83
Capítulo V: Desarrollo del sistema	84
5.1 Módulo de sensado	84
5.2 Aplicación híbrida	85
Referencias	88

Capítulo I: Estudio preliminar

1.1 Introducción

La medición de signos vitales es “un proceso que refleja el estado fisiológico de los órganos vitales: cerebro, corazón, pulmones” [1], es una intervención que forma parte de la valoración del paciente, se realiza de manera constante en Instituciones de Salud.

La toma de signos vitales es una actividad prioritaria y de gran valor para la seguridad del paciente, se hace con el fin de identificar signos y síntomas de alarma en los pacientes, contribuye a disminuir complicaciones, mejorar la satisfacción del usuario y asegurar la calidad de los procesos. Los parámetros que integran la medición son: temperatura corporal, el pulso, la respiración y la presión arterial [1].

Mencionada la importancia sobre la toma de signos vitales se hace notar la necesidad de implementar estrategias de mejora para el proceso de toma de signos vitales, así como el monitoreo sistemático del mismo que asegure la medición y el reporte fidedigno del proceso, contribuyendo a la calidad y seguridad del paciente.

Se recabarán datos generales de los pacientes que serán de ayuda para encontrar atributos semejantes con alguna enfermedad que los pacientes puedan tener o desarrollar, utilizando técnicas de minería de datos, en particular, el árbol de clasificación teniendo como resultado información relevante.

Este sistema puede ser utilizado en el sector salud, tanto en hospitales, como clínicas o centros de salud, en una situación ordinaria e inclusive en una situación extraordinaria. La infraestructura del sistema de salud mexicano ha mostrado tener diversas deficiencias, tanto en la calidad de atención a un paciente, como la escasez de equipo médico o el mal estado y poco mantenimiento de estos, el 24 de diciembre del 2019 el universal informó lo siguiente:

“Todas sus unidades, hospitales y clínicas de primer, segundo y tercer nivel operan con equipos e instrumental médico viejo, que cumplieron su vida útil, lo que pone en riesgo la atención y la salud de 52.7 millones de derechohabientes” [2].

También podría mencionarse la toma de los signos vitales que se llega a hacer con métodos menos precisos y más anticuados, esto podría provocar que el sistema en una situación de extrema delicadeza llegue a colapsar, el ejemplo más reciente de ello fue la llegada de una pandemia, cuando nadie se lo esperaba, en esa ocasión la atención médica en los hospitales fue en extremo demandada, provocando una saturación de los servicios de salud.

En caso de una situación preocupante, como puede ser una pandemia se necesitaría de mayor cantidad de personal dentro de una institución de salud o bien contar con una infraestructura que permita al hospital atender a una mayor cantidad de gente, DOC-CHECKER pretende ser un sistema un tanto más económico que podría ayudar a la toma de signos vitales como la temperatura, pulso cardiaco y la oxigenación o el seguimiento del estado de salud de un paciente con el personal de salud que cuenta un hospital, por supuesto que no sustituye a los enfermeros, pero si ayuda a que un enfermero pueda monitorear más pacientes en alguna situación de gravedad.

Se propone desarrollar un sistema que agilice el tiempo de respuesta de la atención médica, con esto se busca que el sistema sea una ayuda para dotar a los hospitales de tecnología para el monitoreo de estas variables por medio de un sistema accesible económicamente e innovador. La información recolectada será procesada creando un árbol de clasificación con el fin de hallar características generales de los pacientes dentro de algunas enfermedades.

1.2 Objetivo general

- Diseñar un sistema de monitoreo de signos vitales para el seguimiento de pacientes en centros de salud, utilizando sensores conectados a una placa de desarrollo, que enviará sus lecturas a un servidor para su posterior análisis, que permitirá encontrar tendencias en el riesgo de estos mismos y consulta en una aplicación híbrida.

1.2.1 Objetivos específicos

- Desarrollar módulo de sensores utilizando una placa de desarrollo para la recuperación continua de signos vitales.
- Desarrollar el módulo del servidor para realizar el procesamiento, a fin de la identificación de modelos y/o patrones de comportamiento de los datos.
- Desarrollar el módulo de la aplicación híbrida con herramientas para desarrollo web para establecer la comunicación entre la aplicación y los diversos usuarios del sistema.

1.3 Justificación

El monitoreo de los signos vitales de pacientes en los hospitales resulta del uso de técnicas y métodos ineficientes y poco actualizados.

De acuerdo con un diagnóstico del Instituto Mexicano del Seguro Social, fechado en noviembre del 2019, “si se recorre el territorio nacional, los problemas asociados al equipo médico son comunes en unidades médicas e instituciones de atención a la salud, si bien varían en escala de severidad, pero con problemas como: baja calidad en la prestación de los servicios de salud, ocasionada por falta de disponibilidad de equipos médicos involucrados en el diagnóstico y tratamiento del paciente, a causa de no contar con insumos, funcionar inapropiadamente o estar fuera de operación por descompostura o sobreuso” [3].

En cuanto a la infraestructura de mobiliario, el IMSS ha determinado la existencia de equipos médicos e instrumentos quirúrgicos dañados, y en algunos casos se encuentran obsoletos, por haber excedido su vida útil, lo que podría interrumpir el diagnóstico y tratamiento, así como las complicaciones en la enfermedad, provocando un mayor deterioro de la salud del paciente. Imposibilitando a las instituciones brindar un mejor servicio.

“Errores o inadecuadas mediciones en la toma de signos vitales se convierte en un problema de salud pública dado que es la base para el desarrollo estados patológicos en el paciente que pueden llevarle a complicaciones posteriores lo que supone un aumento exponencial del gasto de recursos humanos, económicos, infraestructura para nuestra población.” [4].

La solución pretende ayudar al personal médico a que sus mediciones sean más fiables, por tanto, la propuesta del equipo para ayudar a atacar esta problemática es desarrollar un sistema que se encargue de leer signos vitales del paciente, pulso, temperatura y saturación de oxígeno ya que estos son los principales signos que se recomienda monitorear en un paciente, y que se notifique al Médico encargado de dichos pacientes a través de una aplicación híbrida en caso de que alguno de los signos vitales monitoreados se encuentre en niveles inadecuados.

También se utilizará un software con el que se realizará un análisis de la información obtenida de los pacientes, como: edad, peso, actividad física, sexo, tipo de sangre, lugar de residencia, etc., junto a los signos vitales leídos por el sistema, este análisis arrojará información general con la que agruparemos a los pacientes dentro de una enfermedad que sería propenso a tener, dichas clasificaciones serán algunas enfermedades que afecten principalmente los signos vitales que estemos monitoreando, estas enfermedades se seleccionarán a futuro, de acuerdo con los datos recolectados, se buscará una relación entre los mismos.

Esta propuesta destaca en la portabilidad de datos indispensables para el cuidado del paciente proporcionando herramientas tecnológicas e innovadoras que puedan ayudar al área de la salud a hacer frente a la obsolescencia anteriormente descrita, proporcionando una manera alterna de monitorear el estado del paciente y haciendo frente a una situación adversa, con el propósito de mejorar la infraestructura de los hospitales.

1.4 Productos o Resultados esperados

En la Figura 1 se muestra a grandes rasgos la arquitectura del sistema. El servidor funciona como intermediario entre el módulo de monitoreo y la aplicación del usuario, en este caso, el Médico.

En la Figura 1 se muestra la comunicación del equipo de monitoreo, con el servidor, el cual tendrá que mandar a este mismo los datos leídos para que el servidor los pueda almacenar en una base de datos y enviarlos al usuario de la aplicación, posteriormente se muestra la comunicación entre el servidor y la aplicación del usuario.

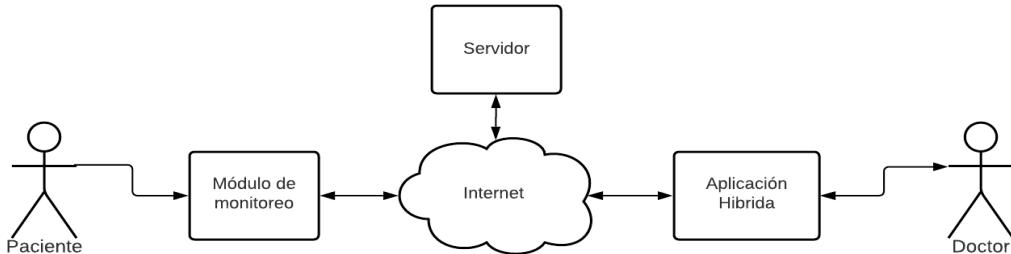


Figura 1.1 Modelo de la arquitectura del sistema

Los productos con los que espera contar el equipo al final del desarrollo del proyecto son los siguientes:

1. Sistema de Monitoreo de Signos Vitales
2. Aplicación Híbrida
3. Módulo de monitoreo
4. Informe basado en el análisis de datos
5. Manual Técnico
6. Manual de usuario

Estas son partes esenciales y que implementan las funcionalidades que se espera tenga el sistema.

1.5 Metodología

Para llevar a cabo el prototipo se consideró como mejor alternativa el uso del modelo en V, puesto que esta metodología es la que mejor se adapta al desarrollo de sistemas embebidos, en ella se parte de un análisis y diseño, sigue con una implementación y concluye con una depuración e integración final, todo esto a través de un proceso que consta de 8 etapas.

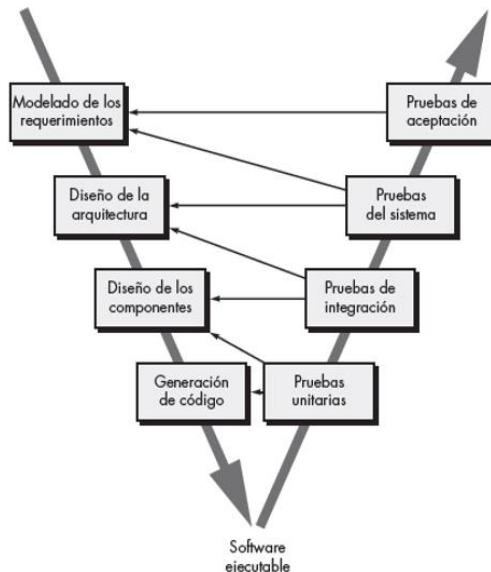


Figura 1.2. Modelo de proceso en V [5].

En el desarrollo del proyecto se pretende especificar el diseño del sistema embebido, el módulo de lectura de signos, la aplicación híbrida y el servidor, así como realizar su implementación. En el test unitario se verifica cada módulo de HW y SW individualmente, depurando cada uno de ellos hasta obtener el resultado esperado. La fase

de integración acopla estos módulos y se concluye con la realización de pruebas sobre un escenario real con el uso del test operacional [6].

Sin duda el utilizar esta metodología para construir Doc-Checker tiene grandes ventajas, como aumentar la calidad del producto, disminuir el tiempo a mercado, tener documentado el proceso, disminuir los rediseños en fase de producción, entre otros factores que redundarán en beneficios para el usuario final.

1.6 Estado del Arte

El desarrollo de productos para la medición de signos vitales es un mercado reducido en que las empresas no se especializan totalmente en la fabricación de estos artículos, por esta razón, no existe una competencia directa, provocando que los productos existentes sean especializados y poco asequibles que imposibilitan compras para dar sustento para todos los pacientes que se necesiten atender en un establecimiento.

Algunos de los sistemas similares que se han desarrollado se muestran en la Tabla 1.1.

Tabla 1.1. Cuadro comparativo de productos similares

Nombre del sistema	Proveedor	Descripción	Alcance	Precio
Monitor de signos vitales PM-7000D [7]	Zoncare	Monitor capaz de medir las variables ECG, HR, NIBP SpO ₂ , 2-TEMP y RESP	Disponible preferentemente para hospitales o individuos con gran valor adquisitivo.	\$31,000
Ritmo Cardiaco App [8]	Azumio Inc.	Aplicación que utiliza la cámara integrada y el flash LED en el teléfono para medir el ritmo cardiaco, genera un gráfico de pulsos por minuto y monitorea una sesión de cardio.	Puede ser utilizado por cualquier persona que cuente con un smartphone que tenga una cámara y un flash.	Gratis
Monitor de signos vitales con comunicación inalámbrica Wi-Fi para unidad de cuidados intensivos desarrollado en LabVIEW y la tarjeta myRIO-1900 [9]	Centro de Investigación de Cómputo	Monitor de signos vitales inalámbrico, mide: -electrocardiografía -oximetría -neumografía -temperatura -frecuencia cardíaca -frecuencia respiratoria -saturación parcial de oxígeno Utilizando la tarjeta myRIO-1900 y envío de datos a través de Wi-Fi desplegando los mismos en una PC con la programación de una interfaz en LabVIEW.	Limitado a participantes del proyecto	Sin comercializar
Xiaomi Mi Watch [10]	Xiaomi	Tamaño de 45,9 x 53,35, pesa 32 gramos Incorpora una pantalla AMOLED de 1,39 pulgadas con una	Disponible para el público	\$749

		<p>definición de 454 x 454 píxeles</p> <p>Batería de 420 mAh</p> <p>Incorpora los siguientes sensores:</p> <ul style="list-style-type: none"> -Sensor cardíaco -Acelerómetro -Giroscopio -Sensor magnético -Barómetro -Luz ambiental -Medición SpO2 -GPS / GLONASS <p>Seguimiento del sueño</p> <p>Bluetooth 5.0</p> <p>Compatible desde Android 4.4 y iOS 10</p>		
Galaxy Watch [11]	Samsung	<p>Tamaño de 46mm</p> <p>Batería de carga rápida de 361 mAh</p> <p>Resistencia al agua IP68</p> <p>Carga inalámbrica</p> <p>Monitoreo de salud</p> <p>Seguimiento diario de tu actividad y estado físico</p> <p>Incorpora los siguientes sensores:</p> <ul style="list-style-type: none"> -BioActive Sensor -Electrocardiograma -Sensor de análisis de impedancia bioeléctrica -SPo2 -Acelerómetro -Barómetro -Giroscopio -Geomagnético -Luz ambiente 	Disponible para el público	\$9,000
Apple Watch [12]	Apple	<p>Mide 40 mm</p> <p>Pesa 30,5 g</p> <p>Su pantalla es de 40 mm con tecnología OLED con una definición de 324 x 394 px</p> <p>Almacenamiento de 32 GB</p> <p>Wi-Fi (802.11b/g/n 2.4GHz)</p> <p>Bluetooth 5.0</p> <p>Cuenta con los siguientes sensores:</p> <ul style="list-style-type: none"> -SpO2 -Ritmo cardiaco -Acelerómetro -Giroscopio -Altímetro 	Disponible para el público	\$30,000

		<p>-GPS/GNSS -Brújula -Sensor de luz ambiental Resistencia al agua de hasta 50 metros La batería tiene una duración de hasta 18 horas</p>		
--	--	---	--	--

Cuadro comparativo con las características principales de los sistemas antes mencionados que se muestran en la Tabla 1.2.

Tabla 1.2. Cuadro comparativo de características principales

Nombre del sistema	Portabilidad	Utiliza una Aplicación móvil	Tiene un registro de los signos	Utiliza Wi-Fi	Utiliza notificaciones push en anomalías del px	Analiza los datos recabados	Costo asequible	Alta precisión
Doc-Checker								
Monitor de signos vitales PM-7000D [7]								
Ritmo Cardíaco App [8]								
Monitor de signos vitales con comunicación inalámbrica Wi-Fi para unidad de cuidados intensivos desarrollado en LabVIEW y la tarjeta myRIO-1900 [9]								

Xiaomi Mi Watch [10]								
Galaxy Watch [11]								
Apple Watch [12]								

El sistema “Doc-Checker” cumplirá con varias características relacionadas con cada uno de estos sistemas, de los principales beneficios identificados con relación a la competencia se encuentran la portabilidad, la experiencia de usuario, la asequibilidad del producto y el análisis de los datos. Al cumplir con estas características nuestro sistema será de gran ayuda para el personal del sector salud mexicano.

Capítulo II: Marco teórico

2.1 Signos vitales y su importancia

Los signos vitales (SV) son valores que permiten estimar la efectividad de la circulación, de la respiración y de las funciones neurológicas y su réplica a diferentes estímulos fisiológicos y patológicos. Los Signos Vitales al ser la cuantificación de acciones fisiológicas como el ritmo cardíaco, frecuencia respiratoria, temperatura corporal, presión arterial y la saturación de oxígeno nos indican que un individuo está vivo, así como la calidad del funcionamiento orgánico. Los signos vitales son útiles en la valoración del dolor postoperatorio en pacientes, a veces cuando están conectados a ventilación mecánica invasiva o cuando inconscientemente, por el contrario, su fiabilidad durante la vigilia depende de la propia explicación del dolor. Los Signos Vitales son una herramienta muy valiosa y de gran ayuda como indicador de un paciente y esta se puede tomar al ingreso y egreso de un paciente a un centro de salud, así como durante su estancia hospitalaria ya sea por turnos o en caso de un estado crítico un monitoreo permanente [13].

2.1.2 Temperatura corporal

La Temperatura Corporal (TC) se comprende como el calor conservado por el equilibrio entre el calor generado (termogénesis) y el perdido (termólisis) por el organismo. El cuerpo activa mecanismos ya sea para subir la temperatura corporal o bajarla para llevarla a una temperatura normal, alteraciones a la temperatura son llamadas hipertermia e hipotermia. La temperatura se mide con un termómetro clínico, en la zona bucal, axilar, ótica, inguinal, rectal o la frente, la mejor forma de tomar la temperatura corporal de cero a 3 meses de edad es rectal, de 3 meses a 3 años puede ser rectal, por oído o en la axila y de 4 años en adelante puede ser tomada oral, por oído y axilar [13].

En la tabla 2.1 se muestra un rango de valores para la temperatura corporal y cómo son denominados en caso de ser demasiado altos o bajos, cómo podrían clasificarse estas distintas temperaturas.

Tabla 2.1. Rangos de la temperatura corporal [14]

Hipotermia muy profunda	Inferior a los 17°C
Hipotermia profunda	entre 17°C a 28°C
Hipotermia ligera	entre 28°C a 35°C
Temperatura normal	entre 36 °C a 37 °C (adultos) 37.5 °C a 37.8 °C (bebés y niños)
Febrícula	entre 37.4 °C a 37.9 °C
Fiebre moderada	entre 38 °C a 38.9 °C
Fiebre alta	entre 39 °C a 39.9 °C
Fiebre muy alta	entre 40 °C a 41.5 °C
Hiperpirexia	> 41.5 °C

La tabla 2.2 contiene rangos de valores adecuados para las personas según su edad. Con estas dos tablas, 2.1 y 2.2, nos podemos percatar de que para la correcta clasificación del valor de la temperatura corporal es tan importante conocer su edad como el valor de esta y determinar una correcta clasificación de la TC de una persona, como valores adecuado o inadecuado.

Tabla 2.2. Por grupos de edad [14]

Recién nacido	entre 36.1 °C a 37.7 °C
Lactante	entre 37.2 °C a 37.8 °C
Niños de 2 a 8 años	entre 37 °C a 37.5 °C
De los 8 a los 15 años	entre 36.5 °C a 37 °C
Adultos (16 a los 65 años)	entre 36.2 °C a 37.2 °C
Vejez (65 años en adelante)	< 36 °C

2.1.3 Frecuencia cardiaca

La frecuencia cardiaca (FC) sufre variaciones normales de acuerdo con la edad, la FC es mayor en los niños y más baja en adultos; a estos le toma más tiempo para que la FC se acelere durante el ejercicio y para que se desacelere al iniciar el reposo [13].

La tabla 2.3 contiene los rangos adecuados de frecuencia cardiaca de una persona de acuerdo con su edad.

Tabla 2.3. Rangos de la frecuencia cardiaca [15]

Edad	Latidos por minuto
Recién nacido	140 - 160
Al año	130 - 115
A los doce años	115 - 100
En el adulto	80 - 70
Anciano	60 - 70

2.1.4 Oximetría

El principio que afirma que la hemoglobina oxigenada y desoxigenada tienen diferente espectro de absorción es la base de la prueba de la Oximetría (OXM), durante esta prueba se emite luz a distintas longitudes de onda, esta luz es transmitida y medida a través de la piel, se calcula la diferencia entre la luz emitida y la luz recibida en cada longitud de onda, indicando el porcentaje de luz absorbida por la sangre, lo que nos permite calcular la saturación de hemoglobina [13].

La oximetría se mide con un aparato de pulsioximetría, por la forma en que es obtenida la lectura se puede obtener a la vez la frecuencia cardiaca, no únicamente el índice de saturación de oxígeno. La técnica para la toma de la oximetría no funciona correctamente si existen uñas pintadas, fuentes de luz externas o anemia. Las zonas idóneas para la toma de la saturación de oxígeno son aquellas partes del cuerpo donde la piel es translúcida y tiene un buen

flujo sanguíneo como los dedos de las manos o de los pies e inclusive el lóbulo de la oreja. El significado de cierto porcentaje de saturación de oxígeno son los siguientes: Mayor del 95% - Pacientes normo saturados; 93% a 95% - Desaturación leve; 88% a 92% - Moderada; Menor al 88% - Grave [13].

La tabla 2.4 contiene las clasificaciones de los valores en la saturación de oxígeno de una persona, considerando los Metros sobre el nivel del mar de la localización en que se esté realizando la lectura de la saturación.

Tabla 2.4. Rangos de la oxigenación en la sangre [16]

Metros sobre el nivel del mar	0 m	1000 m	3000 m	3400 m	3600 m	3900 m
Normal	93-100 %	92-99 %	88-96 %	87-95 %	84-93 %	83-92 %
Hipoxia Leve	88-92 %	88-96 %	87-95 %	83-95 %	80-83 %	79-82 %
Hipoxia Moderada	85-88 %	84-87 %	80-83 %	79-82 %	76-79 %	75-78 %
Hipoxia Severa	<85 %	<83 %	<79 %	<78 %	<75 %	<74 %

2.1.5 Monitoreo de los signos vitales / sensores

El monitor de signos vitales es un dispositivo que detecta, procesa y visualiza continuamente los parámetros fisiológicos del paciente. También tiene un sistema de alarma que puede enviar una alarma en caso de una situación desfavorable o que se supere el límite requerido [17].

El monitoreo continuo es una herramienta muy valiosa para médicos y enfermeras, porque les permite evaluar siempre el estado físico del paciente de forma completa. Además, también puede realizar mejores valoraciones en su tratamiento y diagnóstico. Tomar mejores decisiones [17].

2.2 Servidores

Un servidor es un equipo diseñado para procesar solicitudes y entregar datos a otros ordenadores a los que podríamos llamar clientes. Esto se puede hacer a través de una red local o a través de Internet. Normalmente se configura con capacidad de procesamiento, memoria y espacio de almacenamiento adicional para poder gestionar bien la carga que supone dar servicio a los clientes [18].

2.2.1 Sockets

Un socket es conocido como un tipo de software que actúa como un punto final que funciona estableciendo un enlace de comunicación de red bidireccional entre el extremo del servidor y el programa receptor del cliente. También se le conoce como un punto final en un canal de comunicación bidireccional. Estos sockets se realizan y movilizan junto con un conjunto de peticiones de programación que se identifican como llamadas de función, lo que técnicamente se denomina interfaz de programación de aplicaciones (API). Un socket es capaz de simplificar el funcionamiento de un programa porque los programadores ahora sólo tienen que preocuparse de

manipular las funciones del socket y esto les permite confiar en el sistema operativo para transportar los mensajes a través de la red correctamente [19].

2.3 Internet de las Cosas (IoT, por sus siglas en inglés)

Internet de las cosas (IoT) describe una red de objetos físicos (cosas) que contienen sensores, software y otras tecnologías para conectar e intercambiar datos con otros dispositivos y sistemas a través de Internet. En la actualidad, se tiene la capacidad de conectar a Internet objetos cotidianos (electrodomésticos de cocina, automóviles, termostatos, monitores de vigilancia para bebés, etc.) a través de dispositivos integrados. A su vez, las cosas físicas pueden compartir y recopilar datos con una mínima intervención humana. La comunicación perfecta entre personas, procesos y cosas es posible, en este mundo hiperconectado, los sistemas digitales pueden grabar, supervisar y ajustar cada interacción entre las cosas conectadas [20].

IoT aplicada al sector de la salud actúa de varias formas como en la gestión y recopilación de datos para detectar y prevenir enfermedades, realización de protocolos, ahorro de tiempo y dinero, o monitorización de pacientes y máquinas en tiempo real [20].

2.4 Sistemas Embebidos

Un sistema embebido es un sistema que ha inmerso software y hardware, lo que lo convierte en un sistema dedicado para una aplicación o parte específica de una aplicación o producto o parte de un sistema más grande. Se trata de un dispositivo que incluye una computadora programable pero no pretende ser una de propósito general, es así como una máquina de fax o un reloj con un microprocesador es un sistema embebido.

La computadora es probablemente un microprocesador o un microcontrolador. La palabra embebido implica que esta yace dentro del mismo sistema, oculto a la vista o formando una parte integral de un todo [21].

2.5 Aplicaciones Híbridas

Una aplicación híbrida combina elementos de aplicaciones nativas y aplicaciones web. Las aplicaciones híbridas son esencialmente aplicaciones web que se han colocado en un shell de aplicación nativo. Una vez que se descargan de una tienda de aplicaciones y se instalan localmente, el shell puede conectarse a cualquier capacidad que brinde la plataforma móvil a través de un navegador integrado en la aplicación. El navegador y sus complementos se ejecutan en el back-end y son invisibles para el usuario final [22].

2.6 Bases de datos

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos [23].

2.6.1 Tipos de bases de datos

Existen muchos tipos diferentes de bases de datos. La mejor base de datos para una organización específica depende de cómo pretenda la organización utilizar los datos.

- **Bases de datos relacionales.** Las bases de datos se hicieron predominantes en la década de 1980. Los elementos de una base de datos relacional se organizan como un conjunto de tablas con columnas y filas. La tecnología de bases de datos relacionales proporciona la forma más eficiente y flexible de acceder a información estructurada.
- **Bases de datos orientadas a objetos.** La información de una base de datos orientada a objetos se representa en forma de objetos, como en la programación orientada a objetos.
- **Bases de datos distribuidas.** Una base de datos distribuida consta de dos o más archivos que se encuentran en sitios diferentes. La base de datos puede almacenarse en varios ordenadores, ubicarse en la misma ubicación física o repartirse en diferentes redes.
- **Almacenes de datos.** Un repositorio central de datos, un data warehouse es un tipo de base de datos diseñado específicamente para consultas y análisis rápidos.
- **Bases de datos NoSQL.** Una base de datos NoSQL, o base de datos no relacional, permite almacenar y manipular datos no estructurados y semiestructurados (a diferencia de una base de datos relacional, que define cómo se deben componer todos los datos insertados en la base de datos). Las bases de datos NoSQL se hicieron populares a medida que las aplicaciones web se volvían más comunes y complejas.
- **Bases de datos orientadas a grafos.** Una base de datos orientada a grafos almacena datos relacionados con entidades y las relaciones entre entidades.
- **Bases de datos OLTP.** Una base de datos OLTP es una base de datos rápida y analítica diseñada para que muchos usuarios realicen un gran número de transacciones.

2.7 Minería de Datos

La minería de datos es el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto [24].

Básicamente, la minería de datos surge para intentar ayudar a comprender el contenido de un repositorio de datos. Con este fin, hace uso de prácticas estadísticas y, en algunos casos, de algoritmos de búsqueda próximos a la Inteligencia Artificial y a las redes neuronales [24].

Capítulo III: Análisis de Riesgos

En este capítulo se aborda el análisis de riesgos del proyecto, el cual consiste en el uso de diversas técnicas que sirven como un examen crítico y analítico que tienen como objetivo emitir un juicio y criterio de valor, en respuesta a un peligro determinado. El análisis de riesgos también se define como el uso sistemático de la información disponible para determinar la frecuencia con la que determinados eventos se pueden producir y la magnitud de sus consecuencias [25].

3.1 Identificación de Riesgos

La identificación del peligro consiste en especificar el acontecimiento adverso que es motivo de preocupación. En la tabla 3.1 se enlistan los Riesgos identificados en el proyecto y se asigna un identificador a cada uno de ellos.

Tabla 3.1. Identificación de Riesgos

ID Riesgo	Descripción
R1	Tiempos de las actividades en el cronograma demasiado ajustados.
R2	Requerimientos mal definidos.
R3	Dentro de los integrantes del equipo se tiene diferente disponibilidad de horario.
R4	Poca experiencia en el desarrollo de aplicaciones híbridas.
R5	El software no tiene el uso esperado.
R6	Pérdida de miembro(s) del equipo.
R7	Falta de comunicación del equipo.
R8	Falta de experiencia en el uso de la metodología en V.
R9	Cese de actividades por situaciones extraordinarias.
R10	Fallas en el servicio de red de los miembros del equipo.
R11	Aplicación no intuitiva.
R12	Falla en los servidores donde se aloja la aplicación.
R13	Ineficiencia de hardware para el uso de herramientas de desarrollo.
R14	Pruebas insuficientes en el sistema.
R15	Falta de acceso de parte de los clientes a un servicio de internet.
R16	Mal uso de la aplicación en un ámbito social.
R17	Falta de planificación y seguimiento en la gestión del proyecto.
R18	Confiar en herramientas o tecnologías que no se han manejado antes.

La tabla 3.2 contiene una ponderación asociada a la probabilidad de ocurrencia de un riesgo, estos valores serán utilizados más adelante.

Tabla 3.2. Matriz de probabilidad [25]

Matriz de Probabilidad		
Ocurrencia	Significado	Valor
Frecuente	Casi certeza que se produzca.	5
Probable	Probable que se produzca.	4
Ocasional	Probable que se produzca a veces.	3
Possible	Puede ocurrir en algún momento.	2
Improbable	Nunca puede ocurrir.	1

La tabla 3.3 contiene niveles de gravedad asociados con la ocurrencia de alguno de los riesgos.

Tabla 3.3. Matriz de consecuencia [25]

Matriz de consecuencia		
Consecuencia	Significado	Valor
Catastrófico	Crítico, existen importantes errores, severos incumplimientos al Marco Regulatorio que tiene incidencia en el Banco.	E
Peligroso	Errores significativos continuos, existen incumplimientos a los puntos de control internos y disposiciones legales.	D
Moderado	Errores significativos ocasionales, existen incumplimientos a los puntos de control internos y disposiciones legales.	C
Menor	Errores operativos, existen incumplimientos en algunos puntos de control interno, pero no constituyen infracciones a la Ley.	B
Insignificante	Errores operativos, existen incumplimientos en algunos puntos de control interno que son subsanables inmediatamente.	A

La tabla 3.4 asocia los niveles de probabilidad de ocurrencia de los riesgos con la gravedad de su ocurrencia.

Tabla 3.4. Matriz de probabilidad y consecuencia [25]

		CONSECUENCIA				
PROBABILIDAD		Insignificante A	Menor B	Moderado C	Peligroso D	Catastrófico E
	Frecuente	5				
	Probable	4				
	Ocasional	3				
	Possible	2				
	Improbable	1				

Finalmente se realiza una calificación a cada uno de los riesgos identificados en el proyecto y se coloca el color asociado a los niveles de probabilidad e impacto, los cuales se encuentran en la tabla 3.4. De este modo de forma visual se puede determinar un nivel de importancia para prestar atención a los distintos riesgos, siendo los que están de color rojo aquellos a los que se debe de prestar más atención, pero sin descuidar los demás riesgos.

Tabla 3.5. Probabilidad de Riesgo

Id Riesgo	Probabilidad	Impacto	Semáforo
R1	4	C	Orange
R2	2	E	Orange
R3	3	B	Green
R4	4	D	Red
R5	2	B	Green
R6	2	D	Yellow
R7	3	C	Yellow
R8	4	C	Orange
R9	2	D	Yellow
R10	4	B	Yellow
R11	2	C	Green
R12	4	C	Orange
R13	3	B	Green
R14	3	C	Yellow
R15	2	E	Orange
R16	2	C	Green
R17	3	D	Orange

R18	2	C	
-----	---	---	--

3.1.1 Jerarquización

La jerarquización tiene en cuenta la probabilidad de que se produzca el peligro, las consecuencias si ocurre y el grado de incertidumbre que supone. La tabla 3.6 alberga esta información, de manera similar a la tabla 3.5, pero con la diferencia de que en la tabla 3.6 se realiza un ordenamiento de los riesgos, desde el que tiene un color rojo en el semáforo, hasta el que tiene un color verde.

Tabla 3.6. Jerarquización de Riesgos

Id Riesgo	Probabilidad	Impacto	Semáforo
R4	4	D	Rojo
R1	4	C	Naranja
R2	2	E	Naranja
R8	4	C	Naranja
R12	4	C	Naranja
R15	2	E	Naranja
R17	3	D	Naranja
R6	2	D	Amarillo
R7	3	C	Amarillo
R9	2	D	Amarillo
R10	4	B	Amarillo
R14	3	C	Amarillo
R3	3	B	Verde
R5	2	B	Verde
R11	2	C	Verde
R13	3	B	Verde
R16	2	C	Verde
R18	2	C	Verde

3.1.2 Gestión de riesgos

La gestión del riesgo consiste en la identificación y aplicación de la mejor opción para contener, minimizar o eliminar la probabilidad de que se produzca dicho peligro.

Tabla 3.7. Gestión de Riesgos

Id Riesgo	Descripción	Prevención	Minimización	Contingencia	Semáforo
R4	Poca experiencia en el desarrollo de aplicaciones híbridas.	Estudiar el lenguaje de programación y familiarizarse con las tecnologías que se van a utilizar	Redistribución de tareas según el perfil de los integrantes	Cambiar las tecnologías y herramientas de desarrollo por unas que se conozcan.	Rojo
R1	Tiempos de las actividades en el cronograma demasiado ajustados.	Establecer un cronograma	Reestructuración del cronograma	Ajustar los tiempos limitados para entregar el mayor avance	Naranja
R2	Requerimientos mal definidos.	Hacer una selección de requerimientos obtenidos en una lluvia de ideas	Analizar detalladamente los requerimientos preestablecidos	Adaptar nuevos requerimientos al estado actual del sistema	Naranja
R8	Falta de experiencia en el uso de la metodología en V.	Hacer un estudio exhaustivo de la metodología	Analizar el uso que se le ha dado a la metodología en otros proyectos	Intercalar con una metodología en la que se tenga más experiencia	Naranja
R12	Falla en los servidores donde se aloja la aplicación.	Hacer la elección de un servidor con menor probabilidad de fallas.	Tener conexión con un servidor de respaldo.	Utilizar servidores en la nube.	Naranja
R15	Falta de acceso de parte de los clientes a un servicio de internet.	Se deberá informar que el uso del sistema requiere de Internet	Hacer que el sistema no dependa del todo a una conexión a internet para funcionar	Almacenar las últimas lecturas de los sensores para enviarlas al servidor en cuanto se recupere la conexión de Internet	Naranja
R17	Falta de planificación y seguimiento en la gestión del proyecto.	Apegarnos al cronograma y a las reuniones previamente organizadas	Apresurarse con las tareas que se encuentren en progreso	Dedicar el tiempo necesario a la conclusión de actividades pendientes	Naranja
R6	Pérdida de miembro(s) del equipo.	Mejorar comunicación en	Hablar con los integrantes para	Repartir tareas pendientes entre	Verde

		el equipo.	reestructurar el plan.	los integrantes restantes.	
R7	Falta de comunicación del equipo.	Organizar reuniones en horarios específicos.	Asignar un líder que organice al equipo.	Eliminar las fuentes que limitan la comunicación del equipo.	
R9	Cese de actividades por situaciones extraordinarias.	No depender de algún integrante del equipo	Repartir las actividades del integrante afectado entre los integrantes restantes	Evaluá la disponibilidad para poderle dar prioridad	
R10	Fallas en el servicio de red de los miembros del equipo.	Definir actividades que se puedan desarrollar de forma individual	Desarrollar actividades que se puedan desarrollar de forma individual	Informar a los demás integrantes lo que se realizó durante la desconexión	
R14	Pruebas insuficientes en el sistema.	Pruebas con distintas interfaces	Elegir la interfaz más intuitiva	Actualizaciones de forma continua	
R3	Dentro de los integrantes del equipo se tiene diferente disponibilidad de horario.	Organizar reuniones en horarios específicos	Asignar un líder que organice al equipo	Eliminar las fuentes que limitan la comunicación del equipo	
R5	El software no tiene el uso esperado.	Realizar un diseño del sistema lo suficientemente llamativo	Publicitar el sistema en redes sociales	Valorar un rediseño y actualizar el sistema	
R11	Aplicación no intuitiva.	Realizar pruebas con diferentes interfaces.	Hacer elección de la interfaz más intuitiva.	Valorar la posibilidad de rediseñar la interfaz.	
R13	Ineficiencia de hardware para el uso de herramientas de desarrollo.	Buscar alternativas de software que se ajusten a las especificaciones de nuestro hardware.	Optimizar la forma de trabajo para evitar la sobrecarga del hardware	Usar las instalaciones de la escuela	
R16	Mal uso de la información personal de la aplicación.	Limitar el grado de acceso público de información.	Dejar en claro las restricciones de la aplicación.	Aplicar algún protocolo en caso de filtrado de información.	

R18	Confiar en herramientas o tecnologías que no se han manejado antes.	Descartar herramientas que se desconozcan.	Agregar el aprendizaje de la herramienta o tecnología como una actividad del proyecto.	Cambiar totalmente las tecnologías por unas que se conozcan.	
-----	---	--	--	--	--

3.2 Métricas y Estimación

Para desarrollar un proyecto de software se necesita conocer la cantidad de personas necesarias para llevarlo a cabo y el tiempo necesario del mismo, a pesar de no saber exactamente estos valores si se puede conocer un aproximado, la herramienta que se utiliza para determinar estas aproximaciones son las estimaciones de costos de software. En el desarrollo de proyectos también se necesita realizar estudios y análisis con la intención de mejorar procesos para poder planificar y controlar, en este caso se utilizan las métricas, las cuales nos permiten definir criterios de medición de tamaño [26].

3.2.1 Métrica Puntos de Función

Como métrica orientada a la función se utilizará Punto de Función, la cual es una técnica que nos mide las funcionalidades que ofrece el software desde el punto de vista de sus usuarios [26].

Valores de ajuste de la complejidad:

Tabla 3.8. Ajustes de complejidad

1	¿Requiere el sistema copias de seguridad y recuperación fiables?	5
2	¿Se requiere comunicación de datos?	5
3	¿Existen funciones de procesamiento distribuido?	3
4	¿Es crítico el rendimiento?	5
5	¿Se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?	5
6	¿Requiere el sistema entrada de datos interactiva?	3
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	2
8	¿Se actualizan los archivos maestros de forma interactiva?	5
9	¿Son complejas las entradas, las salidas, los archivos o las peticiones?	5
10	¿Es complejo el procesamiento interno?	5
11	¿Se ha diseñado el código para ser reutilizable?	2
12	¿Están incluidas en el diseño la conversión y la instalación?	1
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	4
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5

Cuenta total (Σf_i)	55
-------------------------------	----

A continuación, se hace un análisis de la cantidad de cada tipo de funciones que realizará el sistema, desde aquellas asociadas con las entradas externas, las salidas externas, peticiones, etc.

Entradas externas

1. Frecuencia cardiaca
2. Oximetría
3. Temperatura
4. Datos del Médico(a) / Enfermero(a)
5. Datos del paciente

Archivos lógicos internos

1. Registros de pacientes
2. Registros de Médicos/enfermeros
3. Actualización de perfiles
4. Registros de signos vitales
5. Análisis de datos

Salidas externas

1. Reporte de análisis de datos
2. Notificaciones al Médico/enfermero
3. Lectura de los signos vitales

Archivos de Interfaz externos

1. Interfaz gráfica de usuario (GUI)
2. Doc-Checker H

Peticiones/Consultas

1. Consultar signos vitales
2. Iniciar sesión
3. Cerrar sesión
4. Consultar reporte

La tabla 3.9 contiene el peso de los tipos de funciones de un proyecto, estos pesos fueron extraídos de [26], los cuales son valores predefinidos. A su vez, para determinar el peso total, se define el peso de los tipos de funciones y se multiplica este valor por la cantidad de funciones de cada tipo.

Tabla 3.9. Parámetros [26]

		Parámetros			
Conteo de funciones		Simple	Medio	Complejo	Total
Nº de Entradas	5	3	4	6	30
Nº de Salidas	3	4	5	7	21
Nº de Peticiones	4	3	4	6	24
Nº de Archivos	5	7	10	15	45
Nº de interfaces externas	2	5	7	10	20
Cuenta total					140

El método de la métrica por puntos de función nos asocia un valor de estos puntos al proyecto, los cuales se obtienen a continuación, con una fórmula extraída de [26], la cual es la 3.1, la cual toma como parámetros los puntos de función obtenido y la cuenta total de la matriz de pesos, de la tabla 3.9:

$$PF = \text{Cuenta Total} \times [0.65 + 0.01 \times \sum f_i] \dots (3.1)$$

$$PF = 140 (0.65 + 0.01 \times 55) = 168 \dots (3.2)$$

Finalmente se determinará un valor estimado para las líneas de código del proyecto o el sistema, las cuales necesitan de un promedio de Líneas de código entre Puntos de función para cada lenguaje de programación (LDC/PF), los valores de la tabla 3.10 fueron extraídos de [26], sin embargo, como se utilizarán distintos lenguajes de programación, se tiene que determinar una media entre estos lenguajes.

Tabla 3.10. Parámetros [26]

Lenguaje de programación	Líneas de Código/Puntos de función (LDC/PF) media
JavaScript	47
Python	50
Promedio total	49

La siguiente fórmula es la que se utilizará para estimar las líneas de código del sistema, en la que se utilizan los Puntos de Función arrojados por esta métrica y el promedio de LDC/PF para los lenguajes JavaScript y Python.

$$LDC = PF \times \left(\frac{LDC}{PF(\text{media})} \right) = (168) \times (49) = 8,232 \dots (3.3)$$

Y ahora si se pueden estimar las miles de líneas de código (KLDC) para el sistema, que es como usualmente se expresa esta estimación.

$$KLDC = \frac{LDC}{1000} = 8.23 \dots (3.4)$$

3.2.2 COCOMO II

El Modelo Constructivo de Costos (COCOMO por sus siglas en inglés que significan Constructive Cost Model), es un modelo de estimación de costos utilizado en los proyectos de Software, es un modelo de formulación matemática, pero con una componente de base empírica. Se basa en estimaciones matemáticas para su formulación, mide el tamaño de un proyecto utilizando como unidad de medida las miles de líneas de código [27]. En la tabla 3.11 se muestran los modos de desarrollo para un proyecto de software y los valores constantes respectivos. Estos valores constantes, codificados como “a”, “b”, “c” y “d”, son propuestos por el modelo COCOMO para complementar las ecuaciones de cálculo usadas en el modelo.

Tabla 3.11. COCOMO II [27]

Modelo COCOMO	Básico			
Proyecto de software	a_b	b_b	c_b	d_b
Orgánico	3.2	1.05	2.5	0.38
Semiacoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.2	2.5	0.32

$$\text{Esfuerzo: } PM = A(KLDC)^B = a_b(KLDC)^{b_b} = 3.2(8.23)^{1.05} = 29.26 \dots (3.5)$$

$$\text{Tiempo de desarrollo: } TDEV = C(PM)^D = c_b(PM)^{d_b} = 2.5(29.26)^{0.38} = 9.01 \dots (3.6)$$

$$\text{Personal} = \frac{\text{Esfuerzo(Persona - mes)}}{\text{Tiempo(mes)}} = \frac{29.26}{9.01} = 3.25 \dots (3.7)$$

El resultado de utilizar el método de estimación COCOMO II nos da un valor de Esfuerzo asociado con el desarrollo del proyecto, el Tiempo estimado de desarrollo y una estimación para el Personal necesario para concluir el desarrollo de este proyecto.

3.2.3 Costo total del Software

En esta sección se realizará una estimación del costo total del proyecto.

$$\begin{aligned} \text{Costo Total} &= \text{Sueldo del participante} * \#Personas * \text{Duración en meses} \\ &\quad + \text{otros costos} \dots (3.8) \end{aligned}$$

$$\text{Otros costos} = \text{Comida, Luz, etc} \dots (3.9)$$

$$\text{Costo Total} = \$11000 * 4 * 5 + \$20000 \dots (3.10)$$

$$\text{Costo Total} = \$240000 \dots (3.11)$$

El costo total contempla el sueldo de los participantes, por lo que para determinar el costo total se debe de conocer el valor total a pagar a los desarrolladores durante este periodo, así como los costos de los servicios básicos [28].

Capítulo IV: Diseño de la solución propuesta

4.1 Análisis de requerimientos del sistema

Se utiliza el análisis de requerimientos para la determinación de especificaciones del sistema, como son sus funcionalidades y las capacidades que tendrá, así como determinar aquello que podría provocar un buen desempeño en la misma y hacer lo más fácil posible la comunicación con el usuario a través de una interfaz gráfica. [29]

4.1.1 Requerimientos funcionales

Los requisitos funcionales nos indican cuál es el comportamiento del sistema ante los estímulos que le llegan del exterior, así como también describen los servicios que presta el sistema. [29]

A continuación, en la tabla 4.1 se mencionan los requisitos funcionales del sistema

Tabla 4.1. Requerimientos Funcionales

ID	Requerimiento Funcional
RF1	El Médico(a) / Enfermero(a) deberán registrarse en el sistema, donde se solicitará información personal (nombre, correo de contacto, cédula profesional, fecha de nacimiento, sexo, teléfono) y tener un perfil.
RF2	El administrador aceptará el registro de los Médico(a) y/o Enfermero(a) al sistema
RF3	El sistema permitirá al administrador cambiar o actualizar información sobre el Médico(a) y/o Enfermero(a).
RF4	El sistema permitirá al Médico(a) y/o Enfermero(a) registrar a sus respectivos pacientes.
RF5	El sistema permitirá al Médico(a) y/o Enfermero(a) cambiar o actualizar información sobre sus pacientes.
RF6	El sistema permitirá al Médico(a) y/o Enfermero(a) observar el estado del pulso, temperatura y saturación de oxígeno del paciente.
RF7	El sistema enviará notificaciones a su dispositivo móvil sobre el estado de los pacientes que estén en gravedad.
RF8	Monitoreo de los signos vitales del paciente
RF9	Consultar el reporte del análisis obtenido por la minería de Datos

4.1.2 Requerimientos no funcionales

Los requisitos no funcionales tienen como objetivo especificar criterios que evalúen la calidad general del sistema, como seguridad, rendimiento, costo, etc., más que especificar el comportamiento del sistema. [30]

A continuación, en la tabla 4.2 se mencionan los requisitos no funcionales del sistema los cuales especifican el comportamiento del sistema frente a algunas situaciones específicas.

Tabla 4.2. Requerimientos No Funcionales

ID	Requerimiento No Funcional
RNF1	Cada Médico puede tener múltiples pacientes
RNF2	Cada paciente puede ser monitoreado por múltiples Médicos
RNF3	Tiempo de respuesta óptimo de parte del sistema

4.2 Análisis de requerimientos Software

A continuación, se realizará la comparación de las alternativas de tecnologías existentes para el desarrollo de Software que podrán ser de utilidad para desarrollar el proyecto, comparación de tecnologías y herramientas que faciliten el desarrollo de Software, pruebas, implementación, etc. Este apartado abarca los distintos componentes del proyecto que tienen una estrecha relación con el Software, como lo es el Servidor, la Aplicación Híbrida, así como también aquello que se utilizará para programar la Placa de Desarrollo.

4.2.1 Análisis de requerimientos de la aplicación híbrida

En el siguiente apartado se buscarán y analizarán distintas alternativas que se puedan utilizar para la elaboración de la aplicación híbrida como la elección del lenguaje de programación, el framework y el entorno de desarrollo a utilizar, se realizarán tablas en las que se tengan las características de las alternativas, sus ventajas y desventajas, con el fin de seleccionar aquellos elementos que mejor se adapten a la elaboración del proyecto.

4.2.1.1 Selección del lenguaje de programación

La tabla 4.3 consiste en un análisis comparativo para seleccionar un lenguaje de programación para el desarrollo de la aplicación, en esta lista tenemos los lenguajes de programación más convenientes para el rubro al que se va a enfocar su uso. Para escoger cada lenguaje candidato también tomamos en cuenta el conocimiento que se tiene de estos y la notoriedad que tienen en este tipo de desarrollo.

Tabla 4.3. Cuadro comparativo de lenguajes de programación para la Aplicación híbrida

Lenguaje	Características	Ventajas	Desventajas
Java [30]	Lenguaje de programación orientado a objetos, fue construido con la intención de escribir una vez el código y ejecutarlo en cualquier dispositivo, esto es posible gracias a su máquina virtual, es un lenguaje que proporciona una plataforma segura para el desarrollo de aplicaciones, puede ser utilizado para el desarrollo web.	Desarrollo más seguro en conectividad y en estabilidad del programa Permite el desarrollo nativo en Android, lo que supone una mejora en el rendimiento de las aplicaciones	Curva de aprendizaje pronunciada No permite el desarrollo híbrido, lo que supondría desarrollar distintas versiones de un programa para ser utilizado en web y en móvil, lo que supone un mayor tiempo de desarrollo
JavaScript [31], [32]	JavaScript es un lenguaje de programación interpretado, es utilizado	Es un lenguaje multiplataforma, lo que permite su ejecución en	Es un lenguaje interpretado, lo que supone un menor

	<p>para el desarrollo de páginas web, utiliza como complemento HTML y CSS, este lenguaje se ejecuta del lado del cliente para crear páginas web dinámicas, lo que permite crear efectos y animaciones con o sin interacción del usuario o bien responder a modificaciones del DOM (Document Object Model).</p>	<p>distintos sistemas operativos Existencia de librerías para realizar cosas más específicas o facilitar la programación como jQuery, React o Angular Es un lenguaje con una curva de aprendizaje baja, por lo tanto, es fácil de aprender Debido a su flexibilidad existen diversas empresas que han desarrollado frameworks para desarrollos con este lenguaje, lo que provoca actualizaciones frecuentes</p>	<p>rendimiento en su ejecución respecto a otros lenguajes Los scripts escritos lo hacen vulnerable ya que cualquier usuario tiene acceso a los códigos</p>
Python [33], [34], [35]	<p>Es un lenguaje de programación orientado a objetos utilizado para el desarrollo web, investigación científica, aprendizaje automático y FinTech, es un lenguaje interpretado con una sintaxis simple y que permite que personas con un menor conocimiento en la materia lo aprendan a utilizar, se ejecuta en sistemas operativos y dominios principales, siendo los sistemas operativos para móviles un reto para este lenguaje, sin embargo, es posible encontrar herramientas y bibliotecas para este fin.</p>	<p>Fácil de aprender por su sintaxis simple</p>	<p>No permite el desarrollo híbrido Falta de herramientas para el desarrollo móvil, únicamente existe el kit de desarrollo Kivy para este fin Problemas de compatibilidad con sistemas operativos para móviles No existe una gran confianza en el desarrollo móvil con Python debido a la complejidad que supone la instalación de las herramientas para la ejecución de código en estos equipos Diversas herramientas para programación móvil se encuentran investigando o en desarrollo</p>

Se utilizará JavaScript para el desarrollo de la aplicación híbrida, ya que tanto Java como Python no se pueden utilizar para este fin, se necesitaría desarrollar tanto una página web como una aplicación móvil utilizando estos lenguajes, lo cual aumentaría el tiempo de desarrollo, por lo que JavaScript es la mejor opción, además de contar con distintos frameworks que facilitan su implementación.

4.2.1.2 Selección del Framework

La tabla 4.4 consiste en una tabla comparativa para seleccionar el framework más conveniente para el desarrollo de la aplicación, en esta lista tenemos los framework más populares y con más impacto. Para realizar una correcta selección de framework se tomaron en cuenta las herramientas con que tiene cada framework orientado hacia el desarrollo de aplicaciones híbridas, así como su curva de aprendizaje.

Tabla 4.4. Cuadro comparativo de Frameworks para la Aplicación híbrida

Framework	Características	Ventajas	Desventajas
Ionic [36]	<p>Permite desarrollar y desplegar aplicaciones híbridas, con una única base de código.</p> <p>Diseño limpio, sencillo y funcional.</p> <p>Emplea Capacitor* para implementar de forma nativa o se ejecuta en el navegador como una aplicación web progresiva.</p> <p>Está construido sobre tecnologías web: HTML, CSS y JavaScript.</p> <p>Se puede usar con los frameworks, como Angular, React y Vue.</p>	<p>Es fácil de aprender y utilizar</p> <p>Numerosas integraciones y plugins</p> <p>Más productividad y menos costos</p> <p>Diseño de interfaces sencillo</p> <p>Buena documentación y respaldo de la comunidad</p>	<p>Peor rendimiento que las aplicaciones nativas</p> <p>Dependencia con los plugins</p> <p>Aplicaciones más pesadas que las nativas</p>
React Native [37], [38]	<p>Compatibilidad Cross-Platform: aplicaciones tanto en iOS como Android con el mismo código base.</p> <p>Funcionalidad nativa.</p> <p>La unión de React Native junto con JavaScript permite la ejecución de aplicaciones más complejas de manera suave</p> <p>Actualizaciones instantáneas.</p> <p>Sencilla curva de aprendizaje.</p> <p>Experiencia positiva para el desarrollador:</p>	<p>Más barato que el desarrollo nativo</p> <p>Acelera el desarrollo y reduce el tiempo de comercialización</p> <p>JavaScript</p> <p>Equipo más pequeño vs. puramente nativo</p> <p>De código abierto</p> <p>Hot reloading (Recarga activa)</p> <p>Gran comunidad de desarrolladores</p> <p>Cercano al rendimiento nativo</p> <p>Cercano a la apariencia y sensación nativas</p> <p>Arquitectura modular</p>	<p>Carece de algunos módulos personalizados.</p> <p>Problemas de compatibilidad y depuración.</p> <p>Se hace necesario tener conocimiento de desarrollo nativo.</p>
Vue [39]	<p>La curva de aprendizaje es la más sencilla de los tres frameworks más populares: React, Vue y Angular.</p> <p>Se trata de un framework muy amigable y respetuoso con las tecnologías de frontend y los estándares. Utiliza HTML, CSS y JavaScript.</p> <p>Se trata de un framework progresivo.</p>	<p>Curva de aprendizaje sencilla.</p> <p>Arquitectura limpia, los componentes quedan mucho más ordenados y limpios.</p> <p>Archivos .vue puedes escribir HTML y css sin tener que hacerlo dentro de JavaScript.</p> <p>Comunidad en crecimiento.</p>	<p>Falta de recursos, algunos aún están en su fase inicial.</p> <p>Riesgo de excesiva flexibilidad. En ocasiones, Vue.js puede tener problemas para integrarse en grandes proyectos.</p> <p>Falta de documentación completa en inglés.</p>

	Vue le da mayor protagonismo al enfoque tradicional «centrado en HTML» así como a los sistemas de plantillas.		
Angular [40], [41]	<p>Angular ofrece la capacidad de crear una single page application, así los cambios son instantáneos no hay que recargar continuamente recargando la página.</p> <p>Genera código optimizado para las máquinas virtuales de JavaScript.</p> <p>Angular es un framework con uso a largo plazo.</p> <p>Permite crear rápidamente vistas de interfaz de usuario con plantillas.</p> <p>Los editores e IDEs obtienen sugerencias de código.</p> <p>Las herramientas de línea de comandos permiten empezar a desarrollar rápidamente, añadir componentes y realizar test, así como previsualizar de forma instantánea la aplicación.</p>	<p>TypeScript, menos errores en aplicaciones al poder usar tipado e interfaces.</p> <p>Framework completo. MCV, datos separados de la lógica y de su representación</p> <p>Inyección de dependencias, que puedes crear mocks de las dependencias.</p> <p>Arquitectura muy bien pensada para proyectos grandes.</p> <p>Servicios, funcionalidad para poder abstraerse de las llamadas a la adquisición de los datos. Puedes tener las llamadas a una API.</p>	<p>Sintaxis compleja heredada de la primera versión de Angular. De todos modos, Angular 5 usa TypeScript 2.4 que es menos difícil de aprender.</p> <p>Pueden aparecer problemas con la migración de anteriores versiones.</p>

El framework que se utilizará será Vue ya que cuenta con la curva de aprendizaje más sencilla entre las otras propuestas de frameworks, además de que se puede usar HTML, CSS y JavaScript.

*Capacitor = Es un framework que permite "transformar" una Web App en una App Mobile o Desktop mediante el uso de WebView.

4.2.1.3 Selección del entorno de desarrollo

La tabla 4.5 consiste en una tabla comparativa para seleccionar el entorno de desarrollo más conveniente para el desarrollo de la aplicación híbrida, en esta lista se tienen los entornos más usados para el desarrollo de aplicaciones. Para escoger el entorno de desarrollo se tomó en cuenta lo intuitivo que sea el entorno, así como la posibilidad de codificación en tiempo real junto con otros colaboradores.

Tabla 4.5. Cuadro comparativo de entornos de desarrollo para la Aplicación híbrida

Plataforma	Características	Ventajas	Desventajas
Android Studio [42], [43]	<p>Un emulador rápido y cargado de funciones</p> <p>Un entorno unificado donde puedes desarrollar para todos los</p>	<p>Es puramente Android, creado para programar en Android.</p> <p>Código más ordenado, estructurado y mejores</p>	<p>Los requisitos son un poco elevados</p> <p>Curva de aprendizaje más lenta para nuevos desarrolladores de</p>

	<p>dispositivos Android</p> <p>Aplicación de cambios para insertar cambios de código y recursos a la app en ejecución sin reiniciarla</p> <p>Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra</p> <p>Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine</p>	<p>sugerencias.</p> <p>Compilador Gradle.</p> <p>Las mejores plantillas para empezar proyectos.</p> <p>Compatibilidad con C++ y NDK</p>	Android.
Visual Studio Code [43], [44], [45], [46]	<p>Tiene una gama extremadamente amplia de utilidades de control de código fuente disponibles a través de extensiones</p> <p>Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código.</p> <p>Posibilidad de configurar la interfaz, se podrá tener más de un código visible al mismo tiempo, las carpetas de nuestro proyecto y también acceso a la terminal</p>	<p>No consume tantos recursos de la computadora</p> <p>Tiene la capacidad de “buscar en cualquier lugar” del proyecto</p> <p>Simplicidad en la interfaz lo que lo hace más intuitivo</p>	No tiene un emulador de Android de forma nativa
Sublime Text [47]	<p>Resalta todo tipo de lenguaje con colores para visualmente detectar fallos a simple vista.</p> <p>Es una buena opción para aprender, porque ayuda, pero no lo da todo hecho.</p>	<p>Muy liviano, fácil de instalar y tiene una versión portable.</p> <p>Funciona tanto en Windows como en Mac y Linux.</p> <p>Hace uso de Monitor para conseguir un ambiente completamente funcional para el desarrollo de Android, sin la molestia de un IDE grande.</p>	<p>No es gratuito, y si no lo tenemos registrado, no pierde ninguna funcionalidad, aunque aparece un mensaje de compra cada cierto tiempo al grabar un archivo.</p>

El entorno de desarrollo Android Studio será el utilizado en el proyecto ya que es el más conveniente para el desarrollo de aplicaciones móviles ya que es un entorno muy intuitivo en donde nos será más sencillo aprovechar cada una de sus herramientas, así como también nos permitirá codificar remotamente a los integrantes del equipo.

4.2.2 Análisis de requerimientos del servidor

En este apartado se buscarán y analizarán distintas alternativas que se puedan utilizar para la elaboración del servidor, como la elección del lenguaje de programación, el Sistema Gestor de Bases de Datos y el entorno de desarrollo a utilizar, se realizarán tablas en las que se tengan las características de las alternativas, sus ventajas y desventajas, con el fin de seleccionar aquellos elementos que mejor se adapten al desarrollo del proyecto.

4.2.2.1 Selección del lenguaje de programación

La tabla 4.6 consiste en una tabla comparativa para seleccionar el lenguaje de programación más conveniente para la elaboración del servidor, en esta lista se tienen los lenguajes más usados para la elaboración de servidores. Para la selección de este lenguaje de programación se tomaron en cuenta algunas necesidades.

Tabla 4.6. Cuadro comparativo de lenguajes de programación para el Servidor

Lenguaje	Características	Ventajas	Desventajas
Java [30]	Java es un lenguaje de programación orientado a objetos	<p>Su plataforma es bastante robusta y estable.</p> <p>Parecido a C++ en estructura y sintaxis, orientado a objetos y con una máquina virtual propia.</p> <p>Java ofrece un mayor nivel de rendimiento con características como la liberación de memoria</p> <p>La gran potencia de su máquina virtual hace que en este aspecto sea la preferida por las empresas</p> <p>Puede usar Spring que es el marco backend de Java más conocido</p>	<p>La arquitectura de Java tiene una gran capacidad de configuración con múltiples opciones, pudiendo llegar a ser muy compleja.</p>
Python [33], [34], [35]	<p>Lenguaje de programación orientado a objetos de alto nivel, que tiene estructuras de datos integradas, combinadas junto a enlaces dinámicos y tipeo, lo que lo convierte en una opción ideal para el desarrollo rápido de aplicaciones.</p> <p>Ofrece soporte para módulos y paquetes, lo que permite la modularidad del sistema y la reutilización de código.</p>	<p>Python permite más flexibilidad en cuanto a la declaración de los tipos de datos, permitiendo que incluso no tengas por qué declararlos.</p> <p>Su simplicidad hace que también se use en la creación de scripts.</p> <p>Capacidad de realizar comprobaciones en el código a la vez que este se ejecuta.</p> <p>Los dos marcos de backend más populares para Python son Django y Flask.</p>	<p>Suele ser más lento que Java en varias implementaciones</p> <p>Curva de aprendizaje</p> <p>La «curva de aprendizaje cuando ya estás en la parte web no es tan sencilla».</p> <p>Hosting</p> <p>La mayoría de los servidores no tienen soporte a Python, y si lo soportan, la configuración es un poco difícil.</p> <p>Librerías incluidas</p> <p>Algunas librerías que trae por defecto no son del gusto de amplio de la comunidad, y optan a usar librerías de terceros.</p>

Node.js [48], [49]	<p>Es un entorno de ejecución de un solo hilo, de código abierto y multiplataforma para crear aplicaciones de red y del lado del servidor rápidas y escalables.</p> <p>Está escrito en C, C++ y JavaScript, y se ejecuta en el motor de ejecución de JavaScript V8.</p>	<p>Node.js utiliza una arquitectura de I/O basada en eventos y sin bloqueos, lo que lo hace adecuado para desarrollar aplicaciones en tiempo real.</p> <p>Puede manejar un número masivo de conexiones simultáneas con un alto rendimiento y proporciona una gran escalabilidad para las aplicaciones.</p>	<p>API Inestable</p> <p>La API de Node tiene la mala costumbre de cambiar en formas que rompen la compatibilidad</p> <p>Falta de una Librería Estándar</p> <p>Falta de Librerías en General</p> <p>Muchas Formas de Programar</p> <p>La falta inherente de organización de código se puede considerar una gran desventaja.</p> <p>No está Probado lo Suficiente</p>
--------------------	---	--	--

Python es el lenguaje de programación que se utilizará para programar el servidor utilizando el framework DJango, sin embargo, al tratarse de una parte fundamental del sistema, éste podría estar sujeto a cambios durante el desarrollo de acuerdo con las necesidades que puedan surgir.

4.2.2.2 Selección del Sistema Gestor de Bases de Datos

En la tabla 4.7 se analizan sistemas gestores de bases de datos que pueden ser útiles para la implementación de la base de datos. La selección de estos es de acuerdo con las necesidades del proyecto y analizamos sus características como el alcance, su costo, el consumo y la experiencia que se tiene en el uso de estos.

Tabla 4.7. Cuadro comparativo de Sistemas Gestores de Bases de Datos

Gestor de bases de datos	Características	Ventajas	Desventajas
--------------------------	-----------------	----------	-------------

MySQL [50]	<p>Sistema de base de datos relacional, lo que quiere decir que archiva datos en tablas separadas en lugar de guardar todos los datos en un gran archivo, lo que le permite tener mayor velocidad y flexibilidad. Estas tablas están relacionadas de formas definidas, por lo que se hace posible combinar distintos datos en varias tablas y conectarlos.</p>	<p>MySQL es de uso libre y gratuito.</p> <p>Software con Licencia GPL.</p> <p>Bajo costo en requerimientos para la elaboración y ejecución del programa.</p> <p>No se necesita disponer de Hardware o Software de alto rendimiento para la ejecución del programa.</p> <p>Velocidad al realizar las operaciones y buen rendimiento.</p> <p>Facilidad de instalación y configuración.</p> <p>Soporte en casi el 100% de los sistemas operativos actuales.</p> <p>Baja probabilidad de corrupción de datos.</p> <p>Entorno con seguridad y encriptación.</p>	<p>Al ser de Software Libre, muchas de las soluciones para las deficiencias del software no están documentadas ni presentan documentación oficial.</p> <p>Muchas de sus utilidades tampoco presentan documentación.</p> <p>Se debe controlar/monitorizar el rendimiento de las aplicaciones en busca de fallos.</p> <p>No es el más intuitivo de los programas que existen actualmente para todos los tipos de desarrollos.</p> <p>No es tan eficaz en aplicaciones que requieran de una constante modificación de escritura en BD.</p>
MariaDB [51]	<p>Se trata de un reemplazo de MySQL que no solo agrega un mayor rendimiento, sino también nuevas funcionalidades.</p> <p>Introduce dos motores de almacenamiento nuevos, uno llamado Aria -que reemplaza con ventajas a MyISAM- y otro llamado XtraDB -en sustitución de InnoDB.</p>	<p>MariaDB Foundation es responsable de la documentación de MariaDB y cuenta para ello con el respaldo de la comunidad de usuarios</p> <p>Incorpora nuevas tablas a nivel del sistema, que ayudan en las tareas de optimización de bases de datos gracias al almacenamiento de estadísticas del servicio.</p> <p>El rendimiento de MariaDB se encuentra por encima del de MySQL.</p> <p>Es compatible con muchos de los scripts más populares utilizados en el mundo</p> <p>Se trata de un proyecto distribuido bajo licencia GPL</p>	<p>Suele tener un retraso al liberar la versión estable, equivalente en nomenclatura a la de MySQL, para poder implementar sus mejoras y realizar las pruebas pertinentes.</p> <p>Existen pequeñas incompatibilidades entre MariaDB y MySQL en algunas versiones (5.1., 5.2)</p>
Oracle [52]	Modelo relacional, visualización de datos en	Motor de base de datos objeto-relacional más	Alto Costo

	<p>tablas con el formato filas/columnas.</p> <p>Herramienta de administración gráfica intuitiva y cómoda.</p> <p>Control de acceso.</p> <p>Protección de datos: seguridad completa en el entorno y gestión de copias de seguridad.</p> <p>Lenguaje de diseño de bases de datos muy completo (PL/SQL).</p> <p>Alta disponibilidad: escalabilidad, protección y alto rendimiento para la actividad empresarial.</p> <p>Gestión de usuarios: agilidad en los trámites, reducción de costos y seguridad en el control de acceso a las aplicaciones y sistemas.</p>	<p>usado a nivel mundial.</p> <p>Multiplataforma: puede ejecutarse desde un PC hasta una supercomputadora.</p> <p>Permite el uso de particiones para hacer consultas, informes, análisis de datos, etc.</p> <p>Soporta todas las funciones que se esperan de un buen servidor.</p> <p>Software del servidor que puede ejecutarse en multitud de sistemas operativos: Linux, Mac, Windows, etc.</p>	<p>Falta de información sobre su uso</p> <p>Incompatibilidad y complejidad</p> <p>Inconsistencia e incompatibilidad de datos en las áreas del tiempo y sintaxis de datos, concatenación de cadenas y sensibilidad de caracteres.</p> <p>Funcionalidad limitada</p>
--	--	--	--

MySQL será el Sistema Gestor de Bases de Datos que se empleará para almacenar la información de usuarios del sistema.

4.2.2.3 Selección del entorno de desarrollo

La tabla 4.8 contiene una comparativa de entornos de desarrollo analizando sus características principales, esto enfocado para llevar a cabo el desarrollo del servidor, una parte fundamental del proyecto y que requiere apoyo de un entorno que se adapte a las necesidades del uso continuo de hardware y software del proyecto.

Tabla 4.8. Cuadro comparativo de entornos de desarrollo para el Servidor

Plataforma	Características	Ventajas	Desventajas
Eclipse [53]	<p>Eclipse es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida con el uso de plug-ins. Es un IDE genérico, compatible con distintos lenguajes de programación.</p> <p>En esta IDE el concepto de trabajo está basado en las perspectivas, que es una preconfiguración de ventanas y editores, que nos permiten trabajar en un determinado entorno de trabajo de forma óptima.</p> <p>Cuenta con una extensa colección de plug-ins, publicados por Eclipse o por terceros, existen tanto</p>	<p>Soporta diversos lenguajes de programación</p> <p>Es ampliable mediante plug-ins</p> <p>Tiene una gran comunidad de Java que lo utiliza</p> <p>Asistentes para la creación de proyectos</p>	<p>No es el más popular entre los desarrolladores Python, existen mejores alternativas</p> <p>Consumo de recursos del sistema</p>

	gratuitos como de pago, bajo distintas licencias		
Visual Studio Code [43], [44], [45], [46]	Editor de texto plano de Microsoft de código abierto y gratuito, ofrece una herramienta avanzada como alternativa de bloc de notas, cuenta con una función de autocompletado, ofrece una amplia variedad de opciones de depuración de código, además de tener soporte para GIT.	No consume tantos recursos de la computadora Tiene la capacidad de “buscar en cualquier lugar” del proyecto Simplicidad en la interfaz lo que lo hace más intuitivo	Pesado. Su base es Electron (Chrome). Prácticamente es como tener un navegador abierto. No tan rápido. Notas un ligero retraso cuando abres archivos, te mueves con el scroll o editas ficheros grandes. No funciona en terminal, aunque si contiene un terminal.
Thonny [54]	Es un IDE de Python, pretende ser sencillo de utilizar para usuarios nuevos en Python, es compatible con Linux, Windows y Mac, ofrece un depurador para corregir errores y permite el acceso a la consola de Python, cuenta con un visualizador de variables y un inspector de objetos.	Diseñado para usuarios poco experimentados en Python Fácil instalación y uso Fácil lectura y depuración de código Puede incorporar plugins	Especializado para Python

Para la programación del servidor se utilizará el IDE Thonny, por sus facilidades y su integración con el lenguaje de programación Python, además, optimizaremos recursos al ser el entorno utilizado también para la programación de la placa.

4.3 Análisis de requerimientos Hardware (módulo de monitoreo)

A continuación, se buscarán y analizarán distintas alternativas que se puedan utilizar para la elaboración del módulo de monitoreo en cada uno de sus apartados como lo es la elección de cada sensor, la elección de la placa de desarrollo, el lenguaje de programación a utilizar en la placa y el entorno de desarrollo para la escritura del código, se realizarán tablas en las que se tengan las características de las alternativas, sus ventajas, desventajas y precio, con el fin de seleccionar aquellos elementos que mejor se adapten a la elaboración del proyecto.

4.3.1 Selección de sensor de temperatura

La tabla 4.9 contiene una comparativa sensores de temperatura analizando sus características principales, esto enfocado para llevar a cabo el desarrollo del sistema de sensado, una parte fundamental del proyecto, para su elección se considera su accesibilidad de compra, características de uso y su precisión.

Tabla 4.9. Cuadro comparativo de sensores de Temperatura

Sensor	Características	Ventajas	Desventajas	Precio
Sensor infrarrojo MLx90614 [55], [56]	Integra un circuito de filtrado de ruido, un convertidor A/D de 17 bits de resolución y un procesador digital de señales Amplio rango de trabajo para objetos desde -70°C hasta 380°C Precisión de 0.5°C Se puede configurar una salida PWM de 10 bits.	El tamaño por su portabilidad Facilidad de uso Buena precisión Suficiente documentación en Internet	Precio Fragilidad Colocación complicada en el paciente	\$280
Sensor de Temperatura RP-RTD [57]	Montaje en superficie Suministro con elemento pasivo Pt100 a 3 hilos. Parche autoadhesivo de goma de silicona Resistente a muchos aceites, líquidos y productos químicos. Adecuado para uso en un rango entre -50 y +150 °C Ideal para su montaje en Placas solares, conductos de calefacción y ventilación o tuberías HVAC. El parche tiene una superficie 40 mm cuadrados Parche en el punto de medida, ofreciendo un aislamiento adicional para el sensor.	Fácil colocación Facilidad de uso	Uso distinto al requerido Precio Precisión más baja que el MLx9014	\$600
ADT7422 [58]	Sensor de temperatura con $\pm 0.1^\circ\text{C}$ de precisión No requiere de ningún circuito externo para su uso Recomendado para uso clínico. Su rango de detección va de los -40 a los 125 °C Utiliza la interfaz serial I ² C, se alimenta con 2.7 a 3.3 V.	Su alta precisión da una ventaja a la hora de medir la temperatura Recomendado para usos clínicos en la medición de signos vitales Facilidad de implementar, ya que no necesita de calibración ni circuitos externos	Dificultad de conseguir Costo algo elevado Únicamente se ha encontrado en tiendas extranjeras	\$425 + envío
MAX30205 [59]	Mide con precisión temperatura y proporciona una alarma de sobretemperatura. La comunicación se realiza a través de un dispositivo de 2 cables compatible con I ² C.	Alta precisión Facilidad de uso Facilita la colocación	Meses en completar la entrega	\$123 + envío

	<p>Este dispositivo está disponible en un paquete TDFN de 8 pines</p> <p>Opera en rango de 0° C a + 50° C.</p> <p>Precisión de 0,1°C (37 ° C a 39 ° C)</p>			
--	--	--	--	--

El sensor de temperatura seleccionado fue el sensor infrarrojo MAX30205 por su precisión, la cual es importante por la naturaleza del proyecto, por sus dimensiones físicas, tamaño y peso, que facilitarán el uso y transporte.

4.3.2 Selección de sensor para oximetría

La tabla 4.10 contiene una comparativa sensores para oximetría analizando sus características principales, esto enfocado para llevar a cabo el desarrollo del sistema de sensado, una parte fundamental del proyecto, para su elección se considera su accesibilidad de compra, características de uso y su precisión.

Tabla 4.10. Cuadro comparativo de sensores para oximetría

Sensor	Características	Ventajas	Desventajas	Precio
Max30102[60]	<p>Integra un pulsioxímetro y monitor de frecuencia cardiaca</p> <p>Compatible con comunicación I2C</p> <p>Se alimenta con un voltaje de 3.3V a 5V, su detección es por reflexión de luz.</p>	<p>Facilidad de uso en el paciente</p> <p>Fácil de implementar</p> <p>Costo asequible</p> <p>Capacidad de medir dos signos vitales</p>	<p>Se requiere un tiempo para estabilizar la correcta lectura de la señal</p> <p>Se utiliza en una zona que tiene mucho movimiento</p>	\$89
MAX32664D [61]	<p>Comunicación I2C</p> <p>Solución de concentrador de sensor biométrico</p> <p>Los algoritmos basados en los dedos miden la frecuencia cardíaca del pulso y la saturación de oxigenación de la sangre del pulso (SpO2)</p> <p>Los datos sin procesar y procesados están disponibles</p> <p>La mezcla periférica básica optimiza el tamaño y el rendimiento.</p>	<p>Precisión</p> <p>Cumple con los requisitos de tamaño, potencia y precisión de los productos sanitarios portátiles</p>	<p>La documentación de este sensor es poco accesible</p> <p>Provee</p> <p>Precio más alto</p> <p>Difícil de conseguir</p>	\$810

El sensor para oximetría seleccionado fue el sensor Max30102 por su doble funcionalidad como sensor de oximetría y de frecuencia cardiaca y su asequibilidad.

4.3.3 Selección de sensor de frecuencia cardiaca

La tabla 4.11 contiene una comparativa de sensores de frecuencia cardiaca analizando sus características principales, esto enfocado para llevar a cabo el desarrollo del sistema de sensado, una parte fundamental del proyecto, para su elección se considera su accesibilidad de compra, características de uso y su precisión.

Tabla 4.11. Cuadro comparativo de sensores de Frecuencia cardiaca

Sensor	Características	Ventajas	Desventajas	Precio
Max30102 [60]	Integra un pulsioxímetro y monitor de frecuencia cardiaca Compatible con comunicación I2C Se alimenta con un voltaje de 3.3V a 5V Su detección es por reflexión de luz.	Facilidad de uso en el paciente Fácil de implementar Costo asequible	Se requiere un tiempo para estabilizar la correcta lectura de la señal Se utiliza en una zona que tiene mucho movimiento	\$89
Ad8232 Ecg Módulo Monitor De Pulso Cardíaco [62]	Proporciona nueve conexiones desde el circuito en el que se puede soldar pines, cables u otras conexiones. Las salidas SDN, LO+, LO-, OUTPUT, 3.3V y GND proporcionan pines para operar el monitor con una tarjeta de desarrollo. Se proporcionan los pines RA (Right Arm), LA (Left Arm) y RL (Right Leg) para conectarlos y usar otros sensores. LED indicador que emite pulsos luminosos al ritmo del bombeo del corazón	Facilidad de uso Electrodos ya creados	Precio elevado Uso muy específico	\$190
Sensor de Pulso Cardíaco [63]	Funciona con un sensor de ritmo cardíaco óptico Una etapa de amplificación y un filtro para el ruido Su señal de salida es confiable y estable. Consumo de corriente bajo siendo de 4 mA con una alimentación de 5 V.	Fácil de configurar para ser utilizado Se encuentra en diversas tiendas en línea	Solo incorpora una de las variables a medir, en comparación con el otro sensor Se utiliza en una zona que tiene mucho movimiento Precio elevado en comparación de otras alternativas	\$189

El sensor de frecuencia cardiaca seleccionado fue el sensor Max30102 por su doble funcionalidad como sensor de oximetría y de frecuencia cardiaca y su asequibilidad.

4.3.4 Selección de la placa de desarrollo

La tabla 4.12 muestra algunas placas de desarrollo que se encuentran en el mercado, estas placas ofrecen distintas prestaciones y se ofertan a distintos precios, cada una tiene características que pueden beneficiar el desarrollo del proyecto, pero para saber exactamente cuáles serían sus ventajas y desventajas en esta tabla se hace una comparativa entre estas distintas alternativas.

Tabla 4.12. Cuadro comparativo de placas de desarrollo

Placa	Características	Ventajas	Desventajas	Precio
ESP32[64]	Procesador: CPU: microprocesador de 32-bit	Bajo costo Documentación	Número limitado de entradas para los	\$146

	Xtensa LX6 de doble núcleo (o de un solo núcleo), operando a 160 o 240 MHz y rindiendo hasta 600 DMIPS Coprocessor de ultra baja energía (ULP) Memoria: 520 KiB SRAM Conectividad inalámbrica: Wi-Fi: 802.11 b/g/n Bluetooth: v4.2 BR/EDR y BLE Interfaces periféricas: 12-bit SAR ADC de hasta 18 canales $2 \times$ 8-bit DACs 10 × sensores de tacto (sensores capacitivos GPIOs) $4 \times$ SPI $2 \times$ interfaces I ² S $2 \times$ interfaces I ² C $3 \times$ UART Controlador host SD/SDIO/CE-ATA/MMC/eMMC Controlador esclavo SDIO/SPI Interfaz Ethernet MAC con DMA dedicado y soporte para el protocolo IEEE 1588 Precision Time Protocol Bus CAN 2.0 Controlador remoto infrarrojo (TX/RX, hasta 8 canales) Motor PWM LED PWM (hasta 16 canales) Sensor de efecto Hall Preamplificador analógico de ultra baja potencia	necesaria para el proyecto	sensores	
Arduino [65]	Es posiblemente la placa más utilizada y robusta. Es una buena opción para iniciarse en la programación de microcontroladores. Estás son sus características técnicas: Microcontrolador: ATMega328P. Velocidad de reloj: 16 MHz. Voltaje de trabajo: 5V. Voltaje de entrada: 7,5 a 12 voltios. Pinout: 14 pines digitales (6 PWM) y 6 pines analógicos. 1 puerto serie por hardware. Memoria: 32 KB Flash (0,5 para bootloader), 2KB RAM y 1KB EEPROM	Documentación extensa Ideal para cualquier lenguaje de programación	Dado que la programación no se realiza en ensamblador, el precio a pagar por el uso de las librerías es un retraso en la ejecución de las instrucciones, algunos microsegundos que en el caso de dispositivos de uso cotidiano son irrelevantes, pero significativos a la hora de hacer adquisición de datos.	\$438-\$820
Raspberry [66]	CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz RAM: 1GB LPDDR2 SDRAM Wi-Fi + Bluetooth: 2.4GHz y 5GHz	Potencia extra para el proyecto. Con el uso se puede producir un	Precio elevado para la estimación del proyecto	\$735-\$1155

	IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps) GPIO de 40 pines HDMI 4 puertos USB 2.0 Puerto CSI para conectar una cámara. Puerto DSI para conectar una pantalla táctil Salida de audio estéreo y vídeo compuesto Micro-SD Power-over-Ethernet (PoE)	calentamiento debido a que no cuenta un sistema integrado de refrigeración.		
--	--	---	--	--

Se seleccionó la placa ESP32, la cual nos provee de las características necesarias para el proyecto, como la potencia, el precio, el tamaño y acepta la comunicación I2C, la cual es de sumo interés para el proyecto, ya que los sensores seleccionados utilizan este tipo de comunicación.

4.3.5 Selección del lenguaje de programación para la placa

Existen diversos lenguajes de programación que se utilizan para programar las placas de desarrollo, desde lenguajes muy sencillos hasta otros más complicados de aprender e implementar, pero también se tiene el factor de la documentación, la comunidad, la cantidad de librerías disponibles para cada alternativa, entre otros factores que podrían afectar la decisión para seleccionar uno u otro lenguaje, por ello en la tabla 4.13 se hace una comparativa entre estos lenguajes de programación que podrían ser ocupados para la programación de la placa de desarrollo.

Tabla 4.13. Cuadro comparativo de lenguajes de programación para la placa de desarrollo

Lenguaje	Características	Ventajas	Desventajas
Java [30]	Es simple Es orientado a objetos Es distribuido Es robusto Es de arquitectura neutral Es seguro Es portable Es interpretado	Es eficiente, rápido y utilizable para situaciones reales. Los requerimientos mínimos son un par de docenas de K para el espacio requerido por el código y un par de cientos de bytes de RAM para empezar.	Tan poco como 512 bytes de memoria de programa Tan poco como 768 bytes de RAM (donde van sus variables. Está limitado a 768 caracteres de cadenas por esta restricción). Alrededor de 20k códigos de operación Java por segundo en 8 MHz AVR. Solo incluye java.lang.Object, java.lang.System, java.io.PrintStream, java.lang.StringBuffer, una clase de control JVM y una clase IO nativa. No podrá importar java.util.*; y obtener todas las clases que no están en esta lista.
MicroPython [67]	MicroPython se ejecuta en varios microcontroladores y no hay grandes limitaciones	Lenguaje de programación sencillo y fácil de aprender. Gran comunidad. Python	Curva de aprendizaje no es sencilla al utilizarlo para microcontroladores

	<p>para trasladar MicroPython a más de microcontroladores siempre que haya suficiente memoria RAM, Flash y potencia de procesamiento para ejecutar el intérprete. Dicho esto, hay varios requisitos clave que un desarrollador debe buscar en un microcontrolador a utilizar para ejecutar MicroPython:</p> <ul style="list-style-type: none"> Un mínimo de 256 Kbytes de flash Un mínimo de 16 Kbytes de RAM Un mínimo de 80 MHz de reloj de la CPU 	<p>es un lenguaje muy popular que tiene una gran comunidad detrás. Lenguaje popular. Python se encuentra actualmente entre los 3 lenguajes más usados hoy en día. Python es un lenguaje de alto nivel lo que hace que el código sea más sencillo y comprensible por los seres humanos.</p>	
C++ [68]	<p>Sintaxis heredada del lenguaje C.</p> <p>Tiene un estándar ISO, conocido como ANSI-C++</p> <p>Lenguaje fuertemente tipado</p>	<p>Es más rápido y eficiente que otros lenguajes de alto nivel usados también para programar microcontroladores (Basic, Java, Python, etc.) y su uso está altamente difundido en aplicaciones profesionales. Con compiladores modernos puede llegar a ser tan rápido como el ensamblador dependiendo de la habilidad y los recursos del programador.</p>	<p>El código escrito en C ocupa más memoria que el escrito en ensamblador, dada una misma aplicación. Sin embargo, los compiladores modernos poseen algoritmos de optimización que logran reducir tremadamente esta diferencia.</p>

Se acordó utilizar el lenguaje de programación micropython por su eficiencia, sencillez en su escritura y compatibilidad con la placa ESP32 contando con una amplia y clara documentación.

4.3.6 Selección del entorno de desarrollo

La tabla 4.14 consiste en un análisis comparativo de entornos de desarrollo enfocados al uso de la placa de desarrollo. Debido a que la placa elegida para desarrollar el proyecto fue la ESP32, se han comparado aquí entornos de desarrollo que son útiles para programar dicha placa.

Tabla 4.14. Cuadro comparativo de entornos de desarrollo para la placa de desarrollo

Plataforma	Características	Ventajas	Desventajas
Arduino IDE [69], [70]	Entorno de desarrollo que se puede utilizar para varios lenguajes de programación, es un entorno multiplataforma que puede ser utilizado en	IDE bastante documentado y con una comunidad lo suficientemente amplia, lo que facilita el uso del entorno	Falta de compatibilidad con algunas placas, que se puede solucionar instalando los drivers del fabricante Para usuarios

	Windows, Mac y Linux, el IDE detecta automáticamente la placa conectada y muestra la memoria FLASH y SRAM ocupada por un proyecto, cuenta con un monitor serie que muestra los datos enviados por la placa programada, puede programar una gran cantidad de placas en el mercado.	Interfaz simple y compacta, lo que facilita el uso del software	experimentados habrá falta de algunas características de otros IDEs
ESP-IDF [71]	Es un software diseñado para utilizar hardware basado en ESP32, ayudando a los desarrolladores a expresar sus ideas utilizando este tipo de hardware para el desarrollo de aplicaciones de IoT utilizando Wi-Fi, Bluetooth, administración de energía, entre otras cosas.	Diseñado específicamente para facilitar el desarrollo con ESP32	Puede resultar difícil de configurar Su comunidad puede ser algo más especializada, lo que incrementa su curva de aprendizaje
Thonny IDE [72]	Es un IDE de Python, pretende ser sencillo de utilizar para usuarios nuevos en Python, es compatible con Linux, Windows y Mac, ofrece un depurador para corregir errores y permite el acceso a la consola de Python, cuenta con un visualizador de variables y un inspector de objetos.	Diseñado para usuarios poco experimentados en Python Fácil instalación y uso Fácil lectura y depuración de código Puede incorporar plugins Permite la programación de microcontroladores ESP32	Compatible únicamente con Python
Visual Studio Code [43], [44], [45], [46]	Editor de texto plano de Microsoft de código abierto y gratuito, ofrece una herramienta avanzada como alternativa de bloc de notas, cuenta con una función de autocompletado, ofrece una amplia variedad de opciones de depuración de código, además de tener soporte para GIT.	Tiene soporte para GIT, el sistema de versionado Se puede emplear para el desarrollo en distintos lenguajes de programación, incluido MicroPython Puede utilizar plugins, lo que lo convierte en un editor modular personalizable Sencillo de utilizar	Sistema pesado No es el más veloz del mercado

El entorno de desarrollo seleccionado fue Thonny IDE por su facilidad de uso y configuración, además de que es un entorno diseñado para facilitar la programación con el lenguaje Python y que es compatible con MicroPython, la desventaja es que no se podrá utilizar GIT.

4.4 Diagrama de Casos de Uso

En el diagrama de casos de uso que se muestra en la figura 4.1, se representarán las funciones del sistema desde el punto de vista del usuario, llamado actor. De este modo, el diagrama de casos de uso muestra la relación entre un actor y sus requisitos, sin ponerlas en un orden lógico.

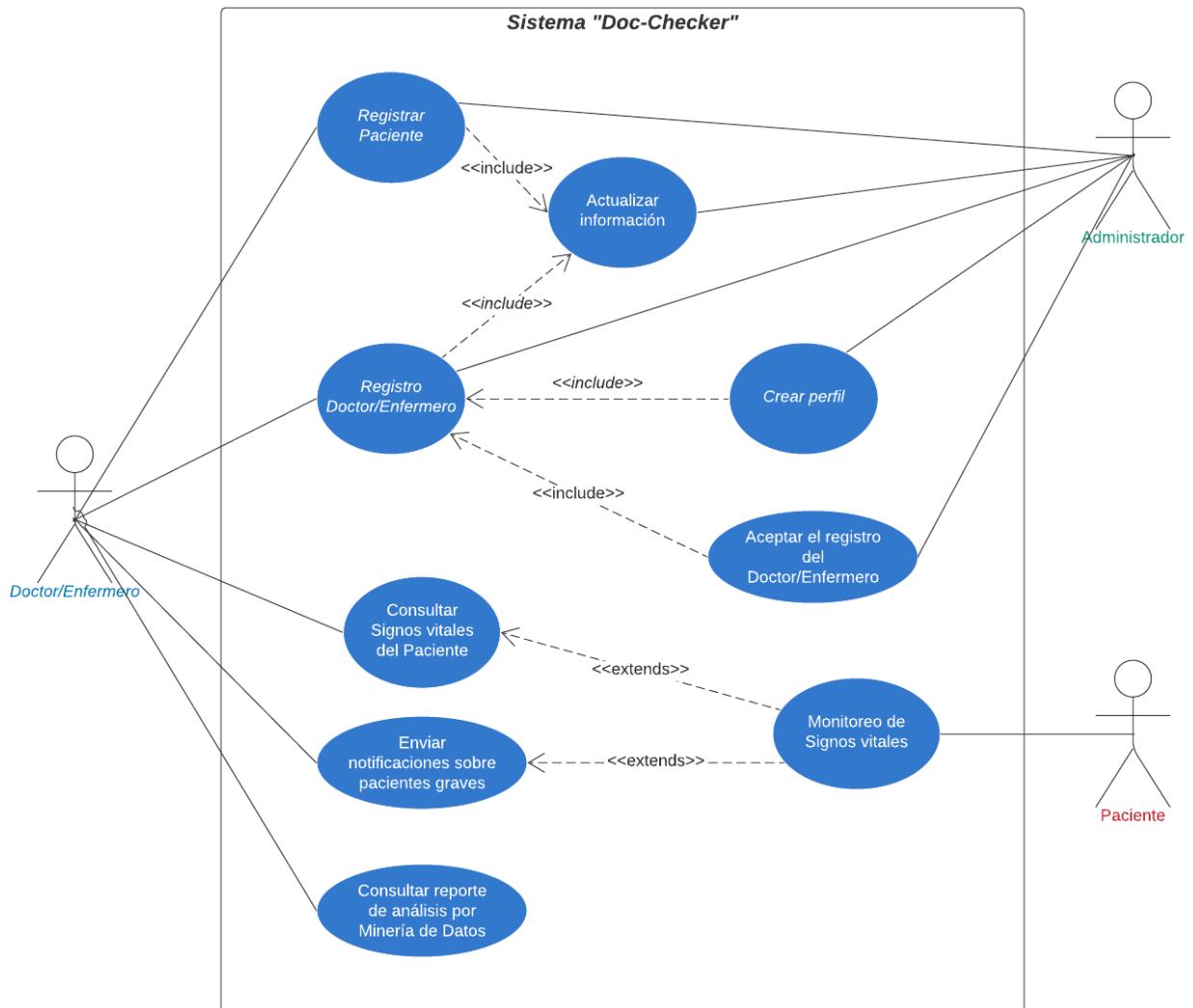


Figura 4.1 Diagrama de casos de uso

4.5 Diagramas de Secuencia

En los diagramas de secuencia muestra la interacción de un conjunto de objetos y se modela para cada caso de uso, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

4.5.1 Diagrama de Secuencia “Registro de Médico/Enfermero”

El diagrama de secuencia correspondiente al registro de un Médico(a) o enfermero(a), el cual se ilustra en la figura 4.2, interactuará directamente con la interfaz del sistema, ya sea en la aplicación móvil o página web la cual verificará en el servidor si ya existe el usuario, si no existe el usuario aceptará el registro y si ya existe devolverá un error, ya que se verificó que el usuario es nuevo y se verifica que los datos son correctos el administrador acepta el nuevo usuario guardando la información en la base de datos.

Diagrama de secuencia: Registro del Doctor/Enfermero

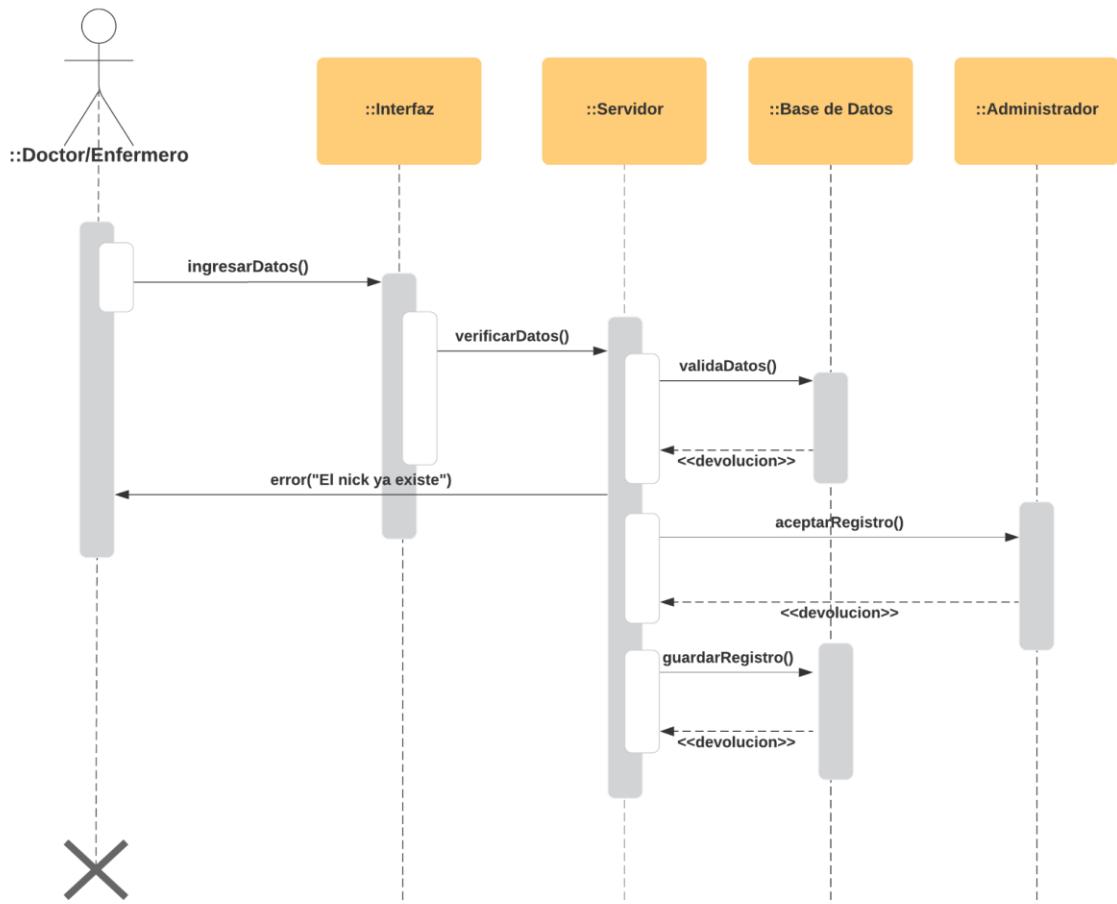


Figura 4.2 Diagrama de Secuencia “Registro de Médico/Enfermero”

4.5.2 Diagrama de Secuencia “Registro de Paciente”

El diagrama de secuencia correspondiente a Registro de paciente, figura 4.3, el médico o enfermero ingresara al paciente por medio de la interfaz, el servidor verifica los datos del paciente con ayuda de la base de datos, después de validar datos, si el paciente ya fue registrado manda un error, si es nuevo el paciente, guardará el registro en la base de datos y el servidor mandará un mensaje de aprobación.

Diagrama de secuencia: Registro del paciente

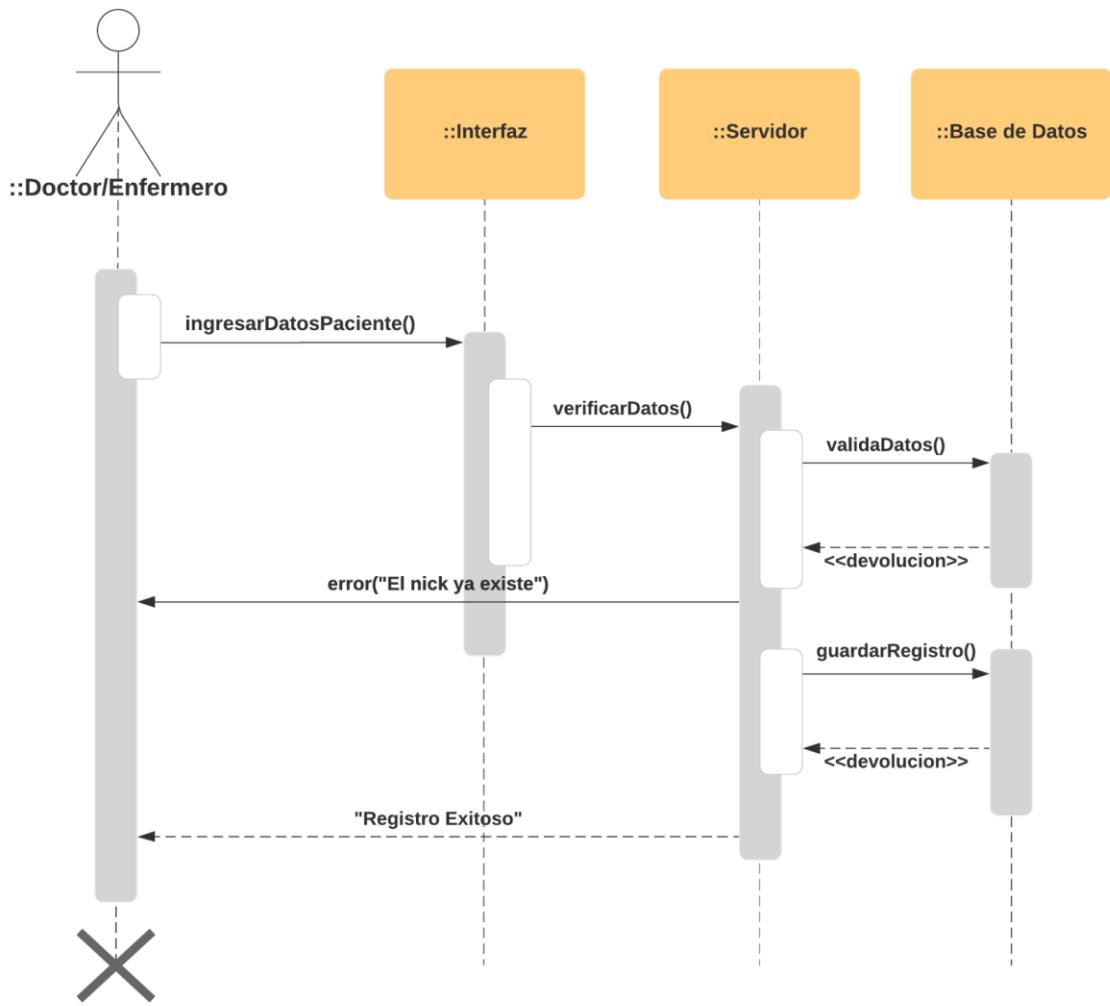


Figura 4.3 Diagrama de Secuencia “Registro de Paciente”

4.5.3 Diagrama de Secuencia “Monitoreo de Signos Vitales”

El diagrama de secuencia de la figura 4.4 corresponde al Monitoreo de los Signos Vitales, el cual incorpora el funcionamiento que se tendrá para esta operación en el sistema, se puede observar que se tiene un paciente del que son leídos sus signos vitales, tanto el servidor como la aplicación podrán recibir información sobre las lecturas de la placa de desarrollo, la placa se encargará de obtener promedios sobre el tiempo especificado por cada cliente y enviará estos promedios a los distintos clientes.

El cliente Servidor almacenará los valores recibidos en la base de datos del sistema para su posterior uso en el análisis con minería de datos, mientras que el cliente Aplicación recibirá los valores que le haga llegar la placa para mostrarlos al usuario de la aplicación híbrida. En caso de que el servidor encuentre alguna anomalía en los valores recibidos con el paso del tiempo se notificará al cliente de la Aplicación híbrida.

Diagrama de secuencia: Monitoreo de Signos Vitales

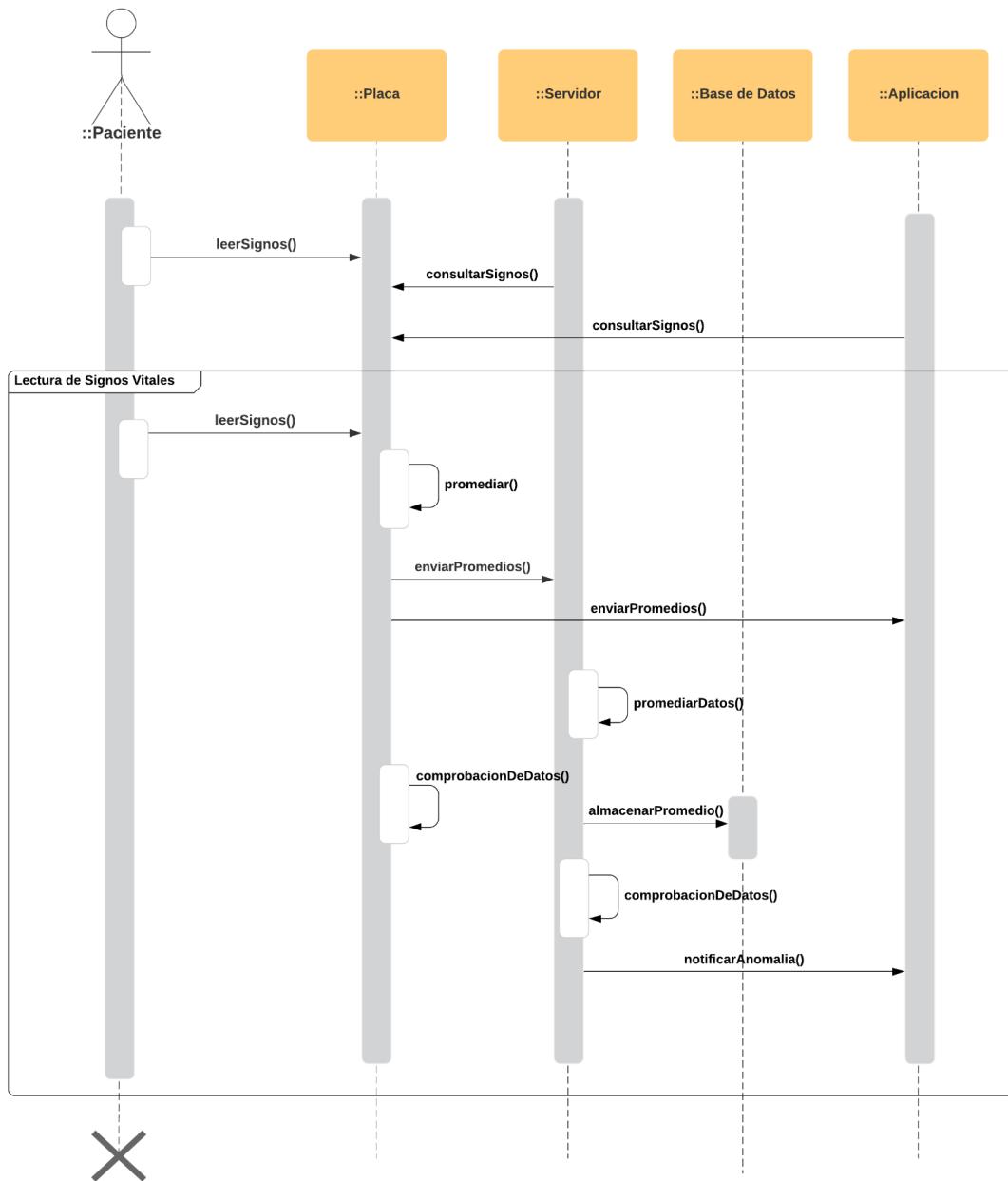


Figura 4.4 Diagrama de Secuencia “Monitoreo de Signos Vitales”

4.5.4 Diagrama de Secuencia “Consultar reporte “Minería de Datos””

El diagrama de secuencia que muestra el funcionamiento de la consulta del reporte de Minería de Datos, ilustrado en la figura 4.5, se representa que el usuario de la aplicación híbrida será capaz de abrir el reporte y visualizar los valores arrojados por el análisis de Minería de Datos, este reporte ya se encontrará generado en el sistema y podrá ser accedido por los distintos usuarios de la aplicación.

Diagrama de secuencia: Consultar reporte "Minería de Datos"

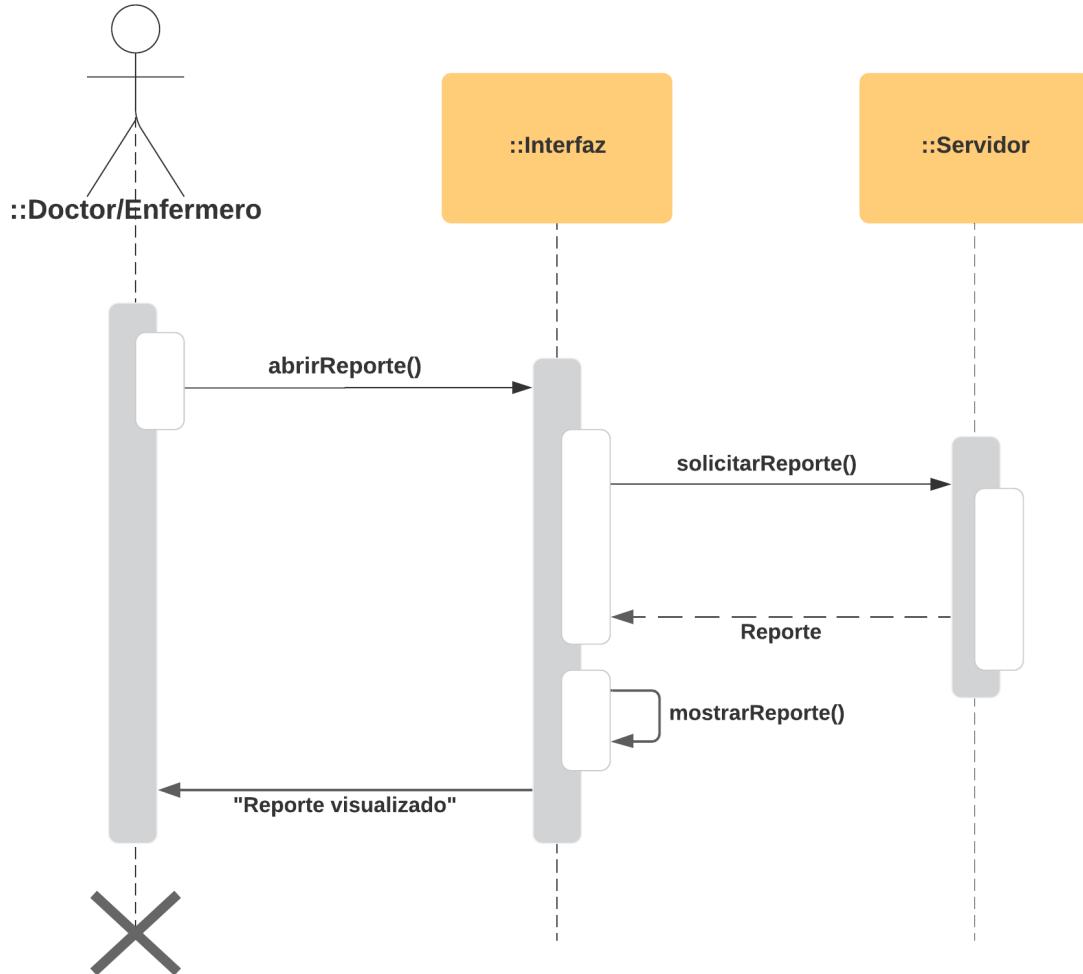


Figura 4.5 Diagrama de Secuencia “Consultar reporte “Minería de Datos””

4.6 Diagramas de Actividades

Los diagramas de actividades mostrarán las secuencias de actividades de un proceso, incluyendo las actividades secuenciales, las actividades paralelas y las decisiones que se toman.

4.6.1 Diagrama de Actividad “Registro de Médico/Enfermero”

El diagrama de actividad de la figura 4.6 muestra de forma global la operación del registro del Médico enfermero, se llenará un formulario que será validado para confirmar que se cuenta con una cédula profesional para poder acceder a los distintos pacientes de algún hospital, posteriormente se agrega el Médico que desea ingresar al sistema a las solicitudes del Administrador del sistema y en caso de ser o no aceptado se enviará un correo electrónico al usuario que intentó realizar su registro.

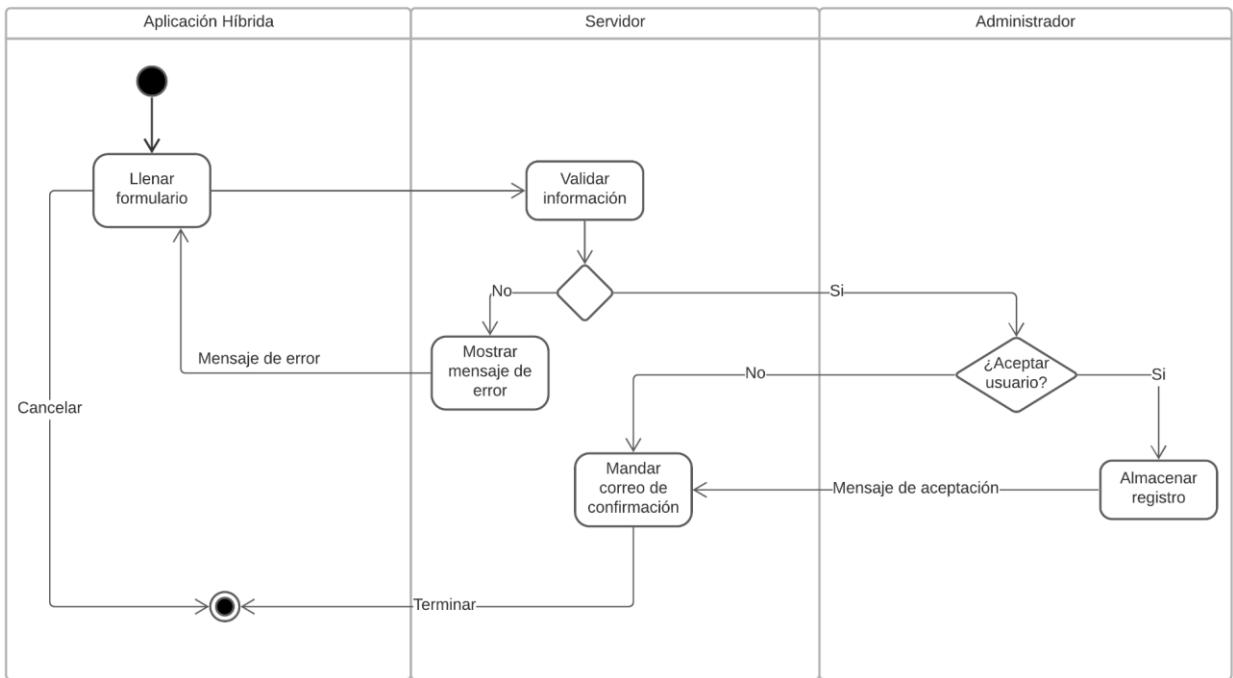


Figura 4.6 Diagrama de Actividad “Registro de Médico/Enfermero”

4.6.2 Diagrama de Actividad “Registro del paciente”

El diagrama de actividad de la figura 4.7 muestra la forma en que se realizará el registro del paciente, iniciando sesión en el sistema y una vez dentro tener la posibilidad de llenar un formulario para la actualización o nueva incorporación de algún paciente al sistema.

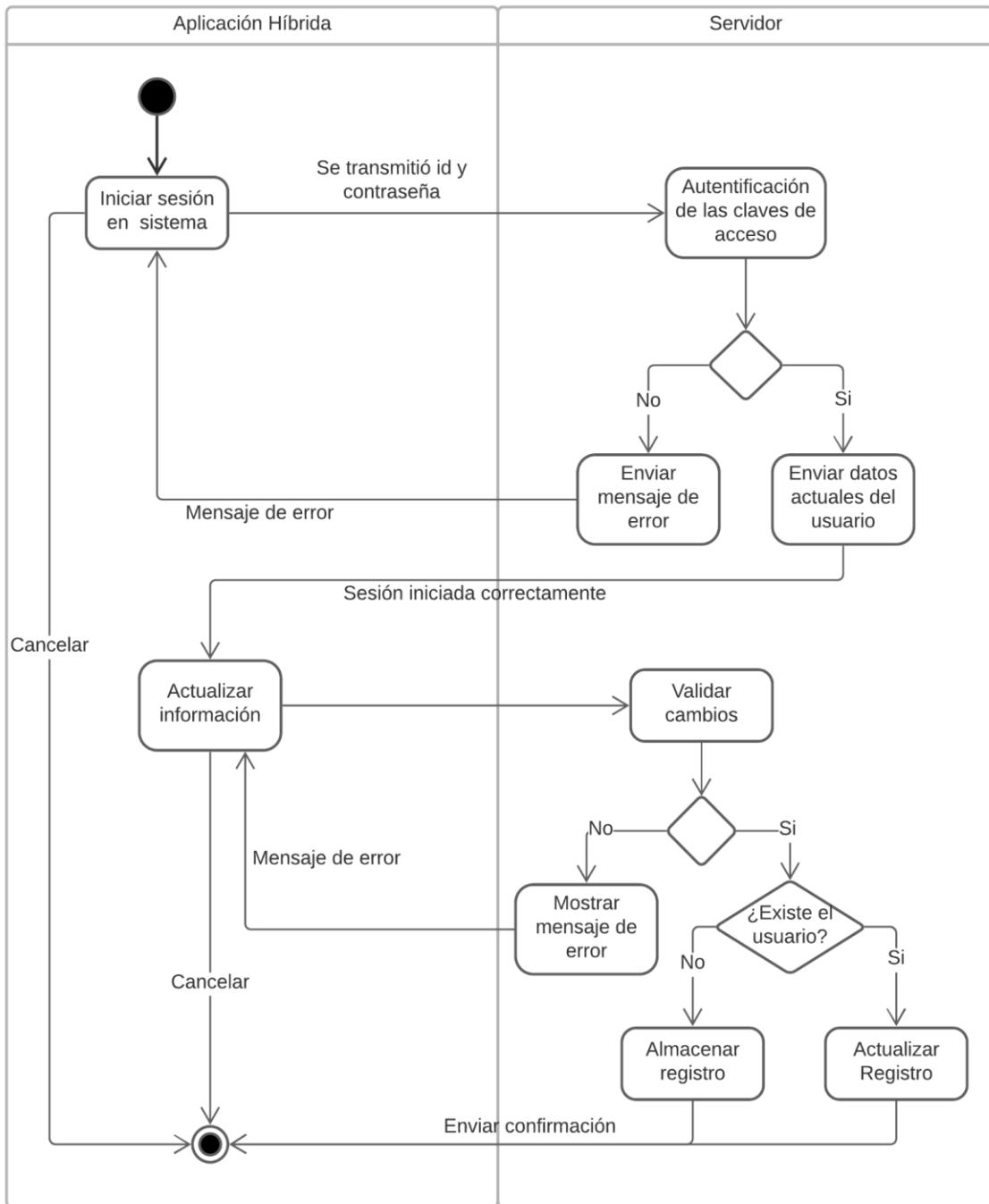


Figura 4.7 Diagrama de Actividad “Registro del paciente”

4.6.3 Diagrama de Actividad “Monitoreo de Signos Vitales”

La figura 4.8 muestra el diagrama de actividad del Monitoreo de Signos Vitales, de forma general se ilustra la lectura de los signos, para posteriormente promediar las lecturas y enviarlas al servidor, en caso de presentar anomalías en las lecturas se enviará una notificación al servidor y posteriormente al usuario encargado del paciente asociado a cierta placa, el usuario podrá abrir la aplicación y consultar las lecturas que se están realizando en ese momento de cierto paciente y podrá simplemente salir de la aplicación o de dicho apartado en la aplicación, informando que ya no desea recibir lecturas en ese momento.

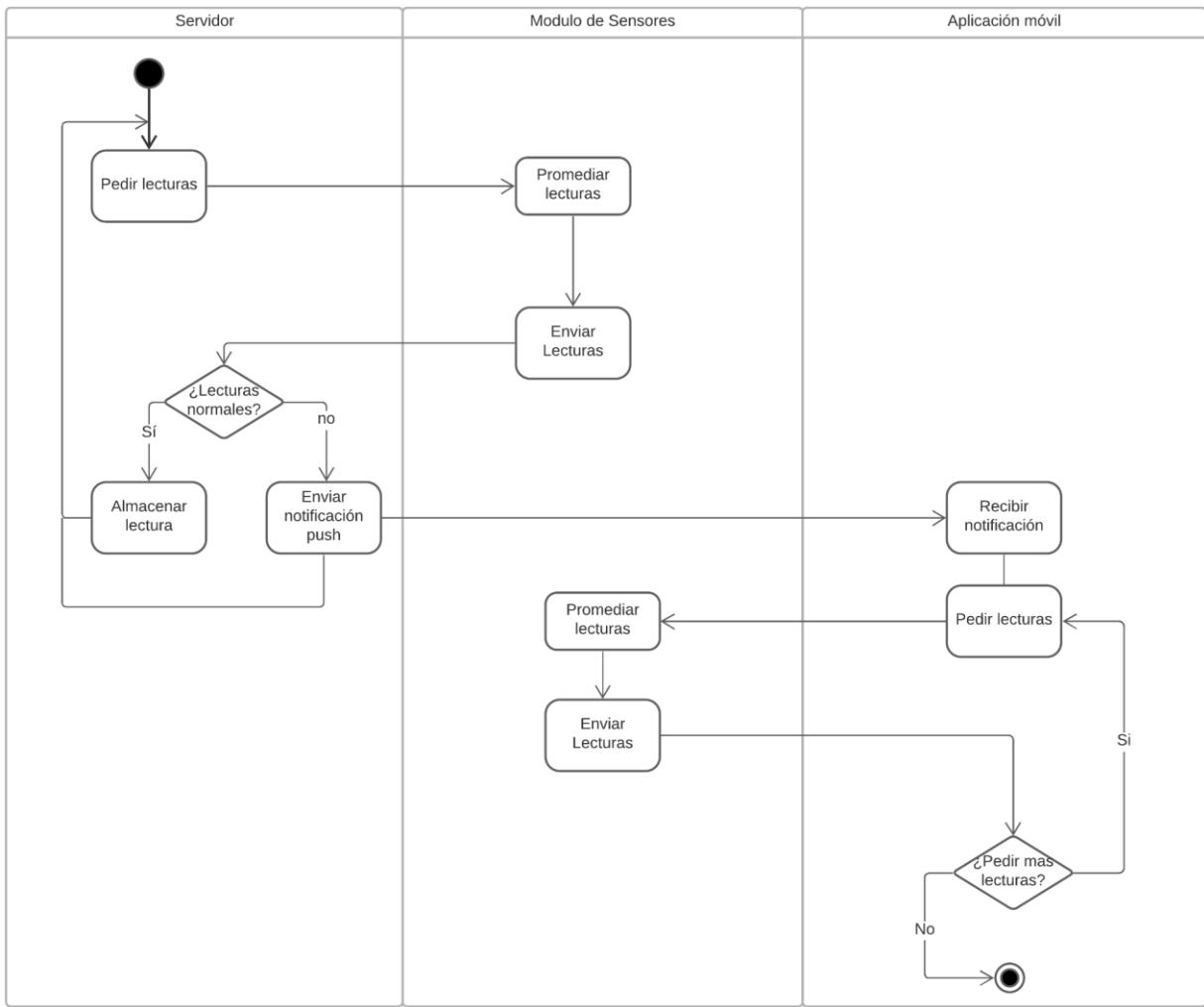


Figura 4.8 Diagrama de Actividad “Monitoreo de Signos Vitales”

4.6.4 Diagrama de Actividad “Consultar reporte “Minería de Datos””

El diagrama de la figura 4.9 muestra las actividades a realizar para la consulta del reporte de Minería de Datos, como solicitar el reporte, recibirla y visualizarlo en caso de que exista, en caso contrario mostrar un mensaje de error.

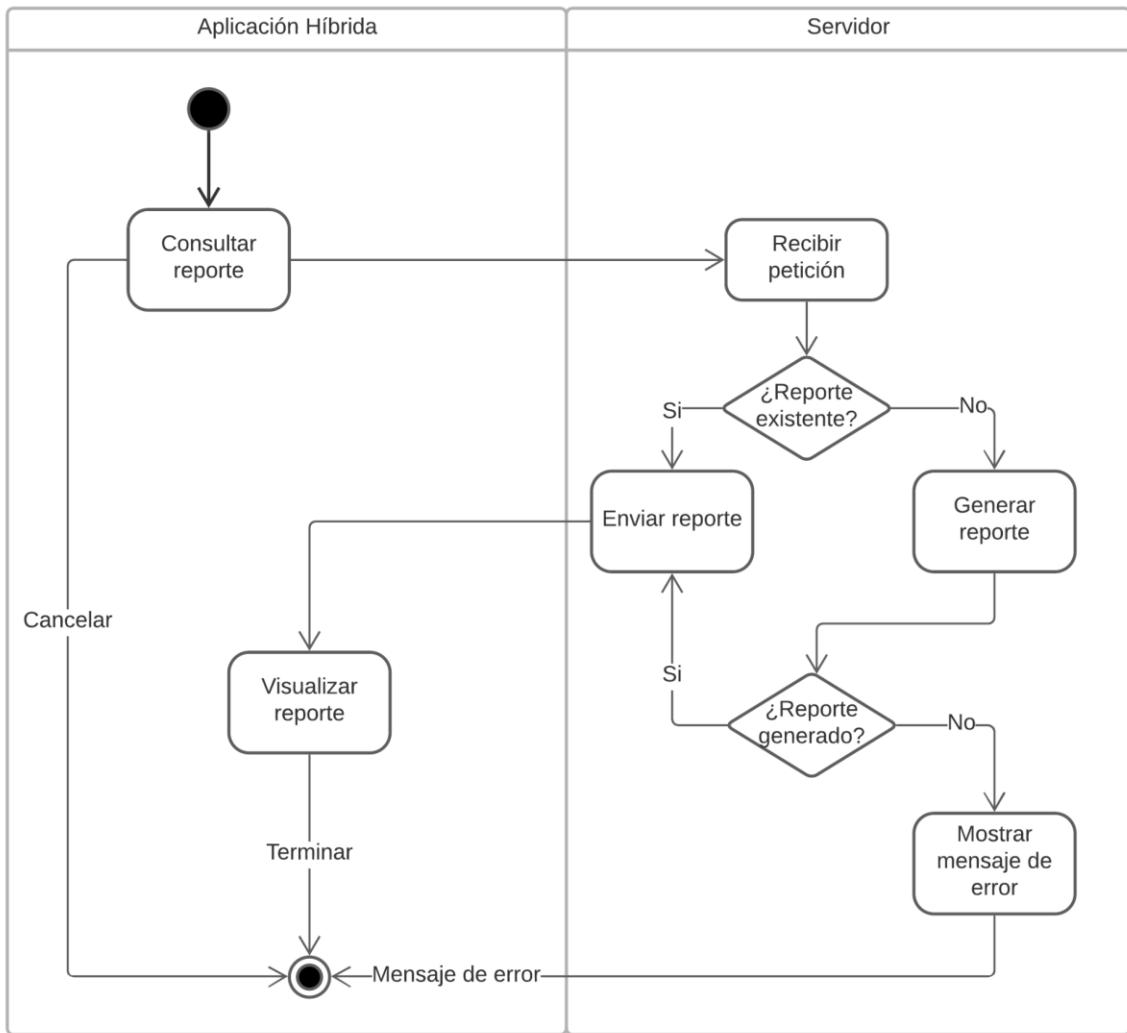


Figura 4.9 Diagrama de Actividad “Consultar reporte “Minería de Datos””

4.7 Diagramas de flujo de datos

Los diagramas de flujo de datos pretenden representar la forma en la que fluirá la información en el sistema, desde qué información se recibe del exterior, hasta cómo será tratada por los distintos procesos y qué se espera obtener de cada uno de esos procesos como salida, esta salida puede entrar nuevamente a otro proceso, puede almacenarse en el sistema para ser utilizada por otros procesos o bien simplemente ser externada del sistema hacia el usuario de este.

4.7.1 Diagrama de contexto

El diagrama de contexto es utilizado para dar una perspectiva global del sistema, sobre aquella información que entrará en él, así como la que saldrá del mismo, en la figura 4.10 se representa el diagrama asociado al sistema desarrollado para este proyecto.

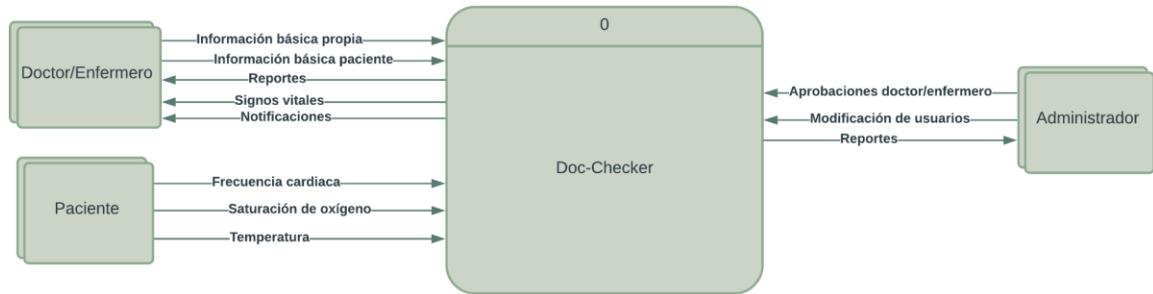


Figura 4.10 Diagrama de contexto

4.7.2 Diagrama de flujo de datos lógico

El diagrama de la figura 4.11 corresponde al diagrama de flujo de datos lógico del sistema que se desarrollará para este proyecto, el cual contiene aquellas funcionalidades principales, la información que recibirá y la información que externará a cada proceso.

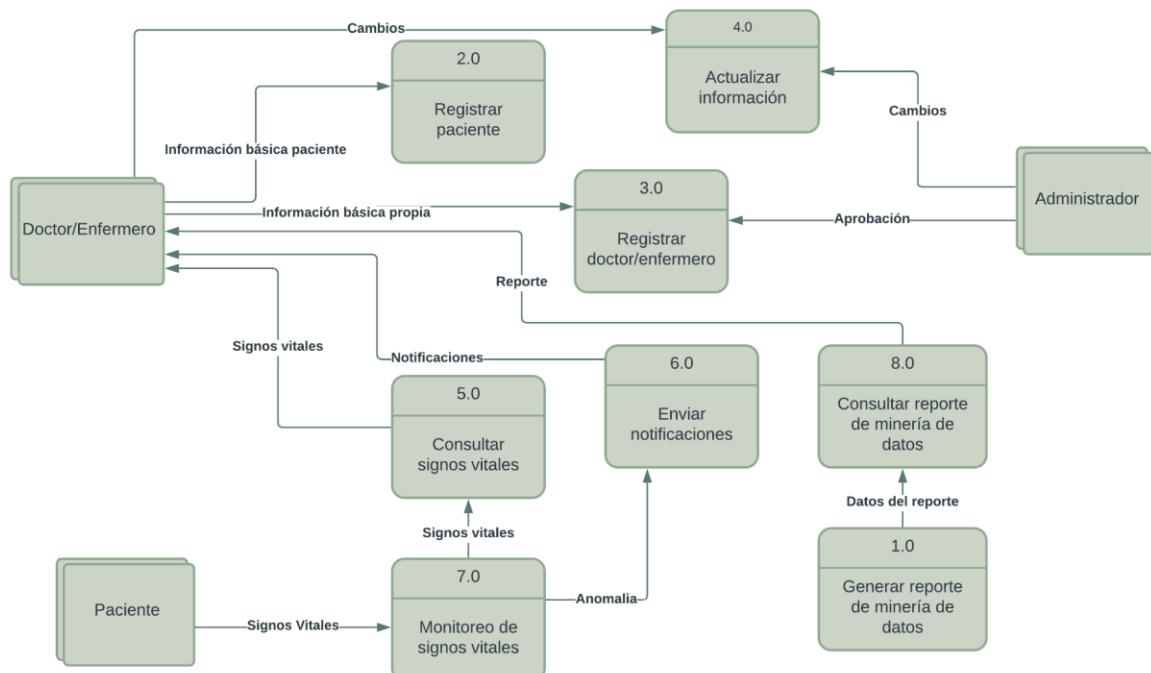


Figura 4.11 Diagrama de flujo de datos lógicos

4.7.3 Diagrama de flujo de datos físico

En la figura 4.12 se tiene el diagrama de flujo de datos físico, bastante similar al diagrama de la figura 4.11, pero aquí se cuenta con la información relativa al almacenamiento de datos, aquella información que se almacenará y aquellos lugares en que se utilizará esa información almacenada, aquí se detalla de una forma más precisa la relación entre cada proceso y se puede ver un funcionamiento más completo del sistema.

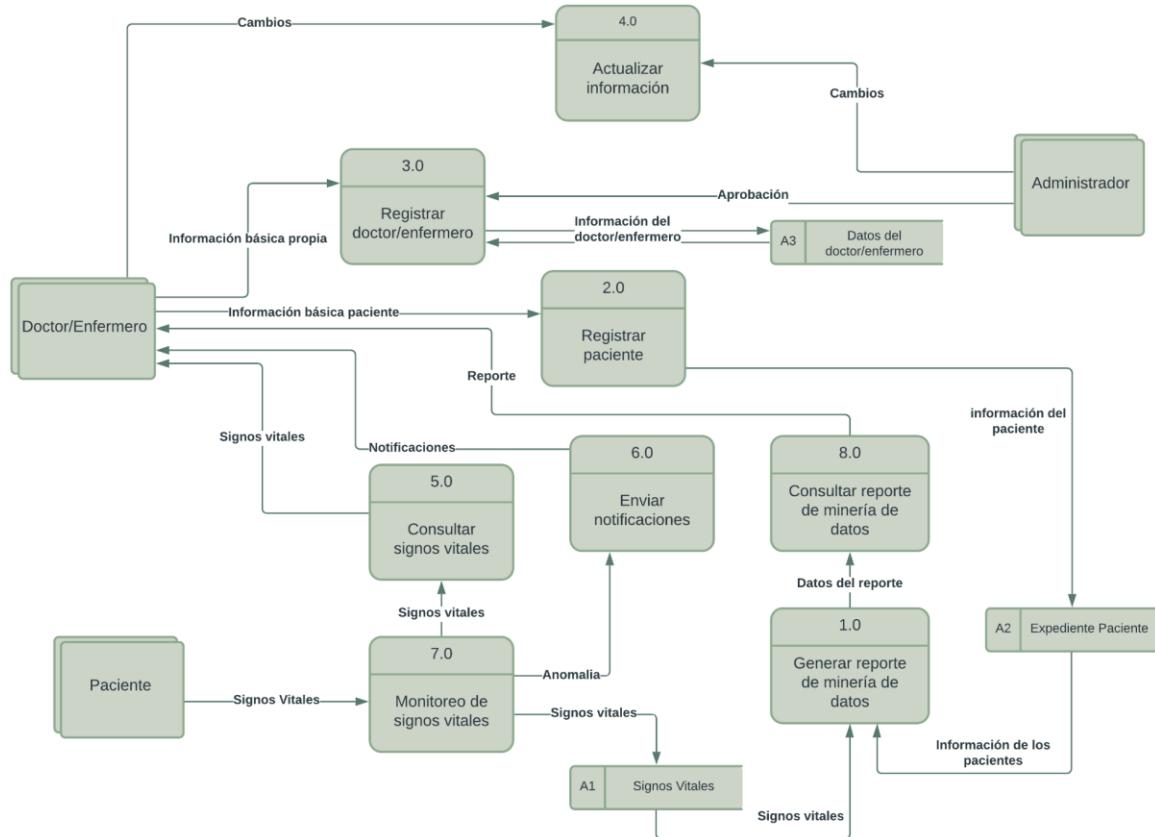


Figura 4.12 Diagrama de flujo de datos físico

4.8 Diseño de la base de datos

La base de datos debe recolectar información relevante que permita realizar un análisis de datos más preciso, estos datos deben ser factores que influyen en la salud de las personas, según la fuente [73], estos factores se clasifican en: ambientales físicos, ambientales sociales, conductuales, biológicos y asistenciales, algunos ejemplos de estos son radiaciones solares, variaciones de temperatura, ruido, pobreza, desempleo, alimentación, adicciones, entre muchos otros. De forma particular para el desarrollo de este proyecto se utilizará el índice de felicidad, temperatura ambiental, humedad, contaminación del aire, del agua, auditiva, la esperanza de vida, la calidad de vida y el bienestar subjetivo, todos estos factores están relacionados a la ciudad en que reside el paciente. También se utilizarán factores más particulares para cada persona, como son padecimientos, antecedentes familiares, exposición solar, variaciones de temperatura, variaciones de humedad, exposición al ruido, actividad física, educación, trabajo actual, horas de sueño, estado civil, consumo de fármacos y saber si tiene personas dependientes.

4.8.1 Diagramas base de datos

Los diagramas de base de datos es una herramienta visual utilizada para construir y organizar bases de datos. Puede mostrar un sistema de base de datos.

4.8.2 Diagrama entidad-relación

El Diagrama Entidad Relación, representado en la figura 4.13, es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades.

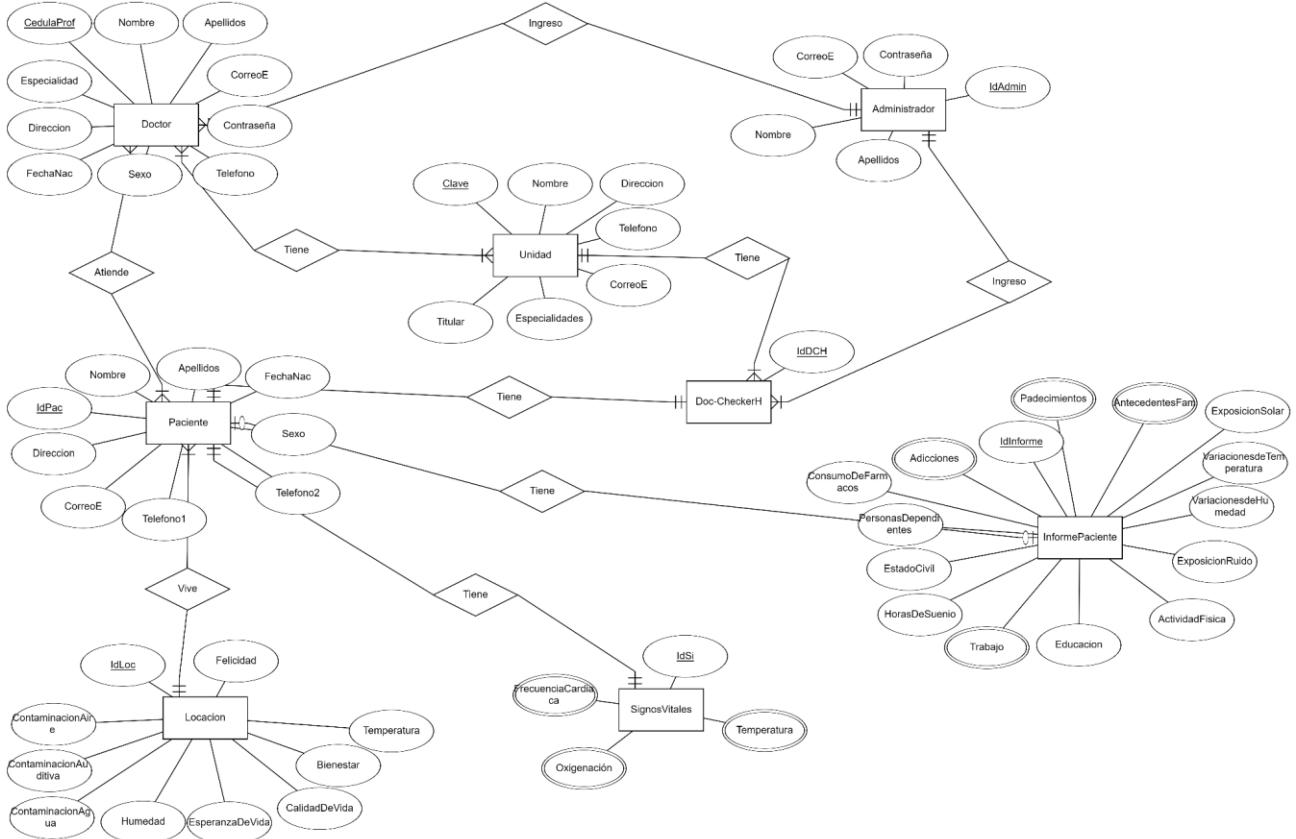


Figura 4.13 Modelo entidad relación

4.8.3 Modelo Relacional

Un modelo relacional consiste en representar datos por medio de tablas relacionadas, la figura 4.14 presenta el modelo relacional diseñado para la base de datos a utilizar en el proyecto.

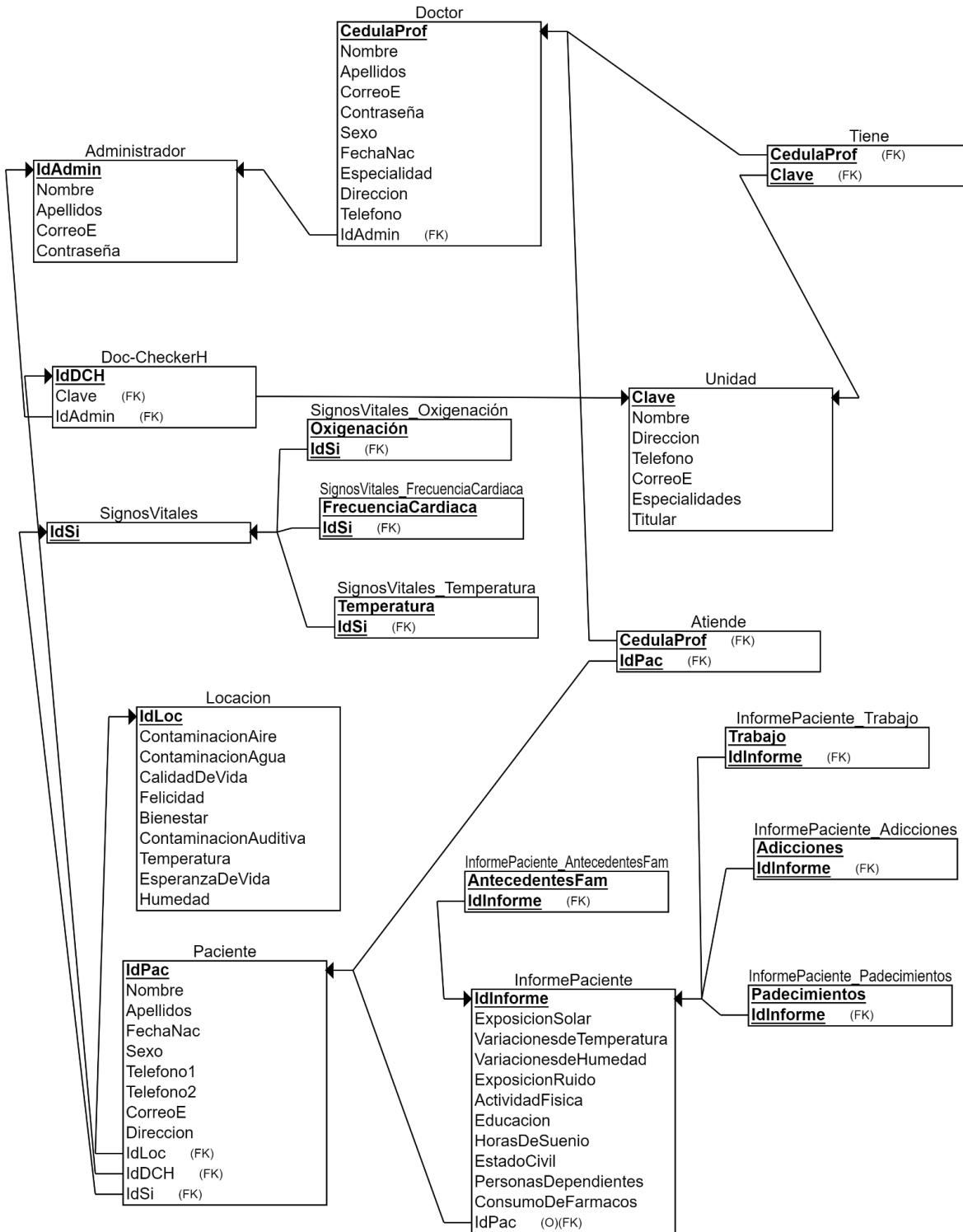


Figura 4.14. Modelo relacional

4.9 Comunicación de los componentes

Los componentes principales que se tienen en el sistema y que deben de comunicarse entre sí son los sensores, la placa de desarrollo, el servidor y la aplicación híbrida, de forma general esto es representado en el diagrama de la figura 4.15. La forma en que se plantea que sea esta comunicación es en primer lugar que los sensores se comuniquen con la placa de desarrollo haciendo uso del protocolo de comunicación I2C, de este modo las lecturas de los sensores podrán ser conocidas por la placa de desarrollo, una vez que se tiene la información de los signos

vitales en este punto se debe de hacer llegar la información hasta el servidor o la aplicación híbrida, puesto que se seleccionó una placa de desarrollo que contará con la tecnología Wi-Fi esta comunicación se puede realizar a través del protocolo TCP/IP, y así conseguir navegar la información a través de Internet o bien en una red a nivel local, esta última forma será la empleada por cuestiones de desarrollo del sistema y seguridad. Así pues se tiene que la lectura de los signos vitales podrá llegar hasta el servidor teniendo el conocimiento de la dirección IP asignada a la placa de desarrollo y a través del servidor se podrá dar a conocer esta IP hacia cualquier usuario de la aplicación híbrida, con lo que se podrá realizar una conexión directa entre la placa de desarrollo y la aplicación híbrida, de este modo se reduce el tráfico de información entre la placa, el servidor y la aplicación híbrida a solo la placa y la aplicación híbrida. De esta parte se concluye que en primer lugar cada usuario de la aplicación híbrida tiene que establecer una comunicación con el Servidor y finalmente se podrá establecer una conexión con la placa de desarrollo de forma directa.

Para el envío de notificaciones a la aplicación híbrida la placa informará de la anomalía detectada y se informará al servidor, el cual sabrá que hay una relación entre la placa, el paciente y su encargado, por lo que de esta forma se sabrá a qué usuario del sistema enviar la notificación. Las anomalías también podrán ser identificadas en el servidor, por lo que desde este punto será enviada la notificación.

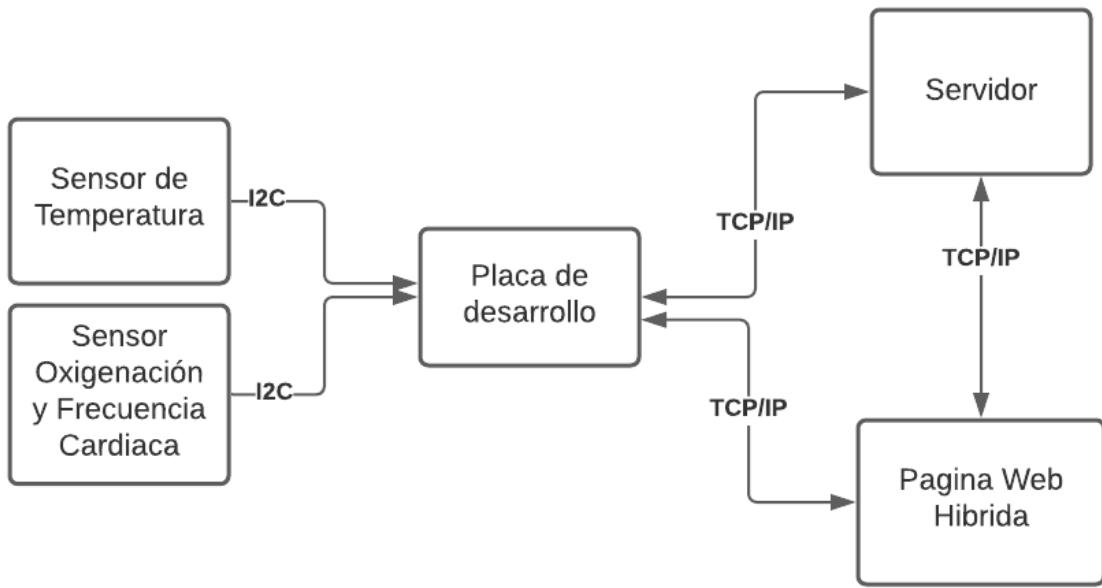


Figura 4.15. Modelo relacional

4.10 Mockups de la Aplicación Híbrida

En las siguientes figuras se muestran los Mockups de nuestra aplicación híbrida, los Mockups son un maquetado de diseño digital para visualizar un prototipo del proyecto que realizaremos.

4.10.1 Inicio de Sesión

El mockup de la figura 4.16.1 corresponde al inicio de sesión, aquella interfaz gráfica con la que se encontrará el usuario en primera instancia, en donde se solicita el usuario y la contraseña, así como también se encontrará el botón para realizar el registro en el sistema.

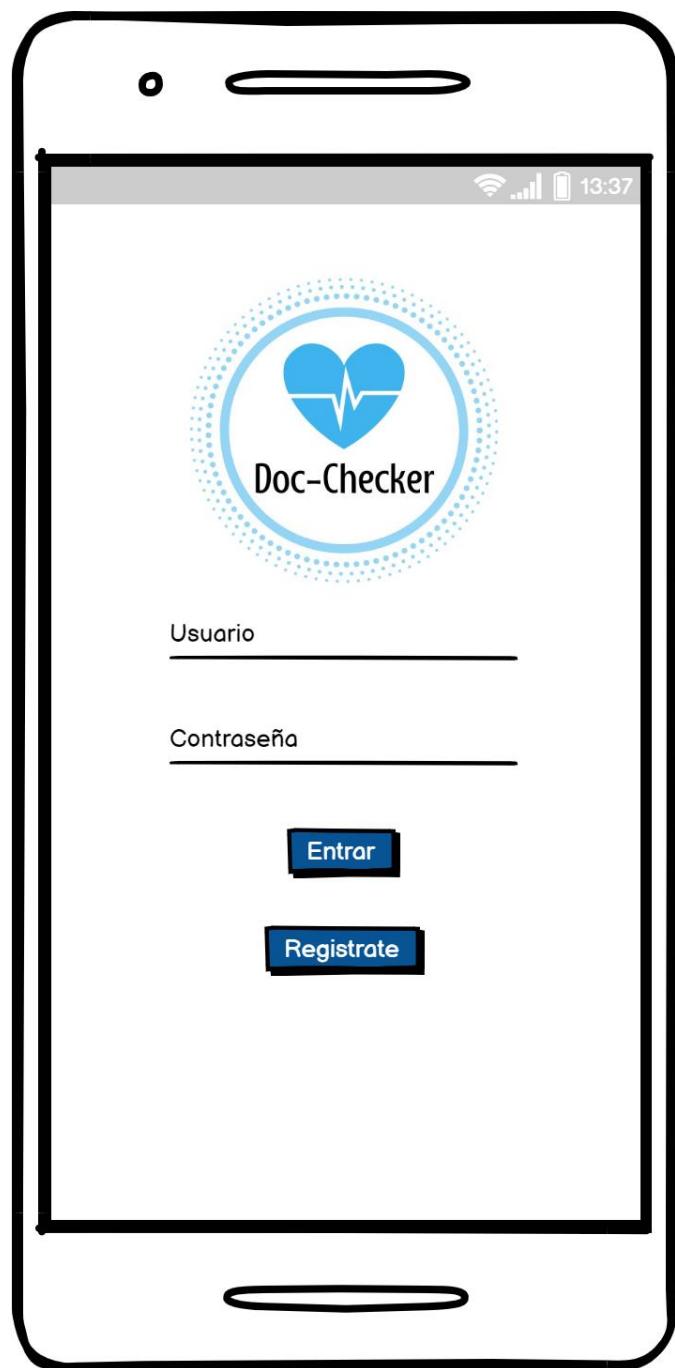


Figura 4.16.1. Inicio de sesión (móvil)

La figura 4.16.2 corresponde de igual forma con el inicio de sesión, pero en esta ocasión es el mockup de la versión web del sistema.

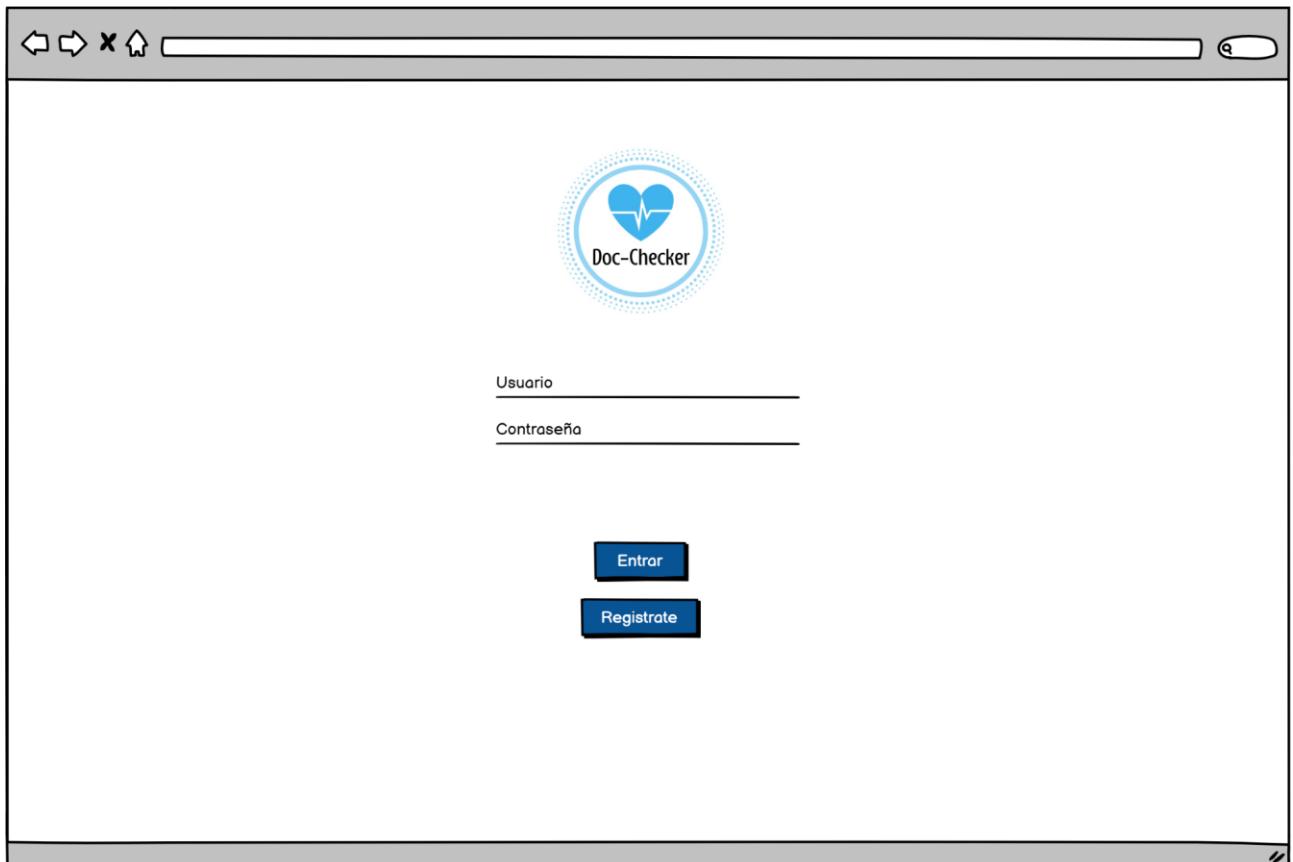


Figura 4.16.2. Inicio de sesión (web)

4.10.2 Registro

La figura 4.16.3 es el mockup del registro del Médico o enfermero, en donde observamos que se solicita información básica de contacto y también validar si el usuario que se desea registrar es médico de profesión, lo que involucraría que necesita de ingresar su cédula profesional para su posterior evaluación por parte del administrador del sistema, esta imagen corresponde al mockup de un dispositivo móvil.

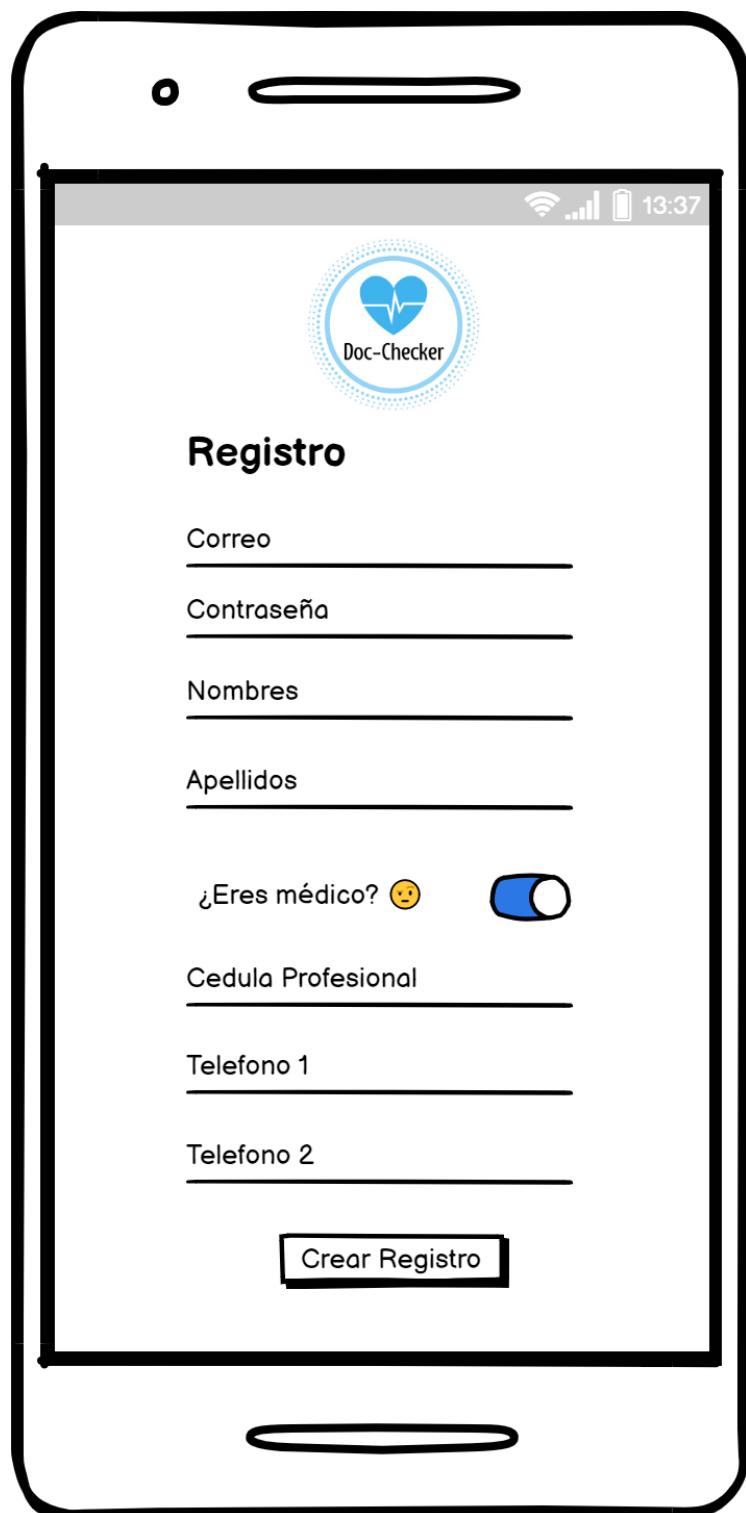


Figura 4.16.3. Registro de nuevo usuario (móvil)

La figura 4.16.4 es la versión web del Registro del Médico/enfermero.

The screenshot shows a web browser window with a light gray header bar containing standard navigation icons (back, forward, stop, refresh) and a search bar. Below the header is the logo for 'Doc-Checker', which features a blue heart icon with a white ECG line inside, surrounded by a circular pattern of small dots. The main content area is titled 'Registro' (Registration). It contains several input fields: 'Correo' (Email), 'Contraseña' (Password), 'Nombres' (Names), and 'Apellidos' (Last Name). Below these are two rows of input fields: 'Cedula Profesional' (Professional ID) and 'Telefono 1' (Phone 1), followed by 'Telefono 2' (Phone 2). To the left of the 'Cedula Profesional' field is a question '¿Eres médico?' (Are you a doctor?) with a yellow smiley face icon. To the right of this question is a blue toggle switch that is currently turned on. At the bottom center of the form is a black rectangular button labeled 'Crear Registro' (Create Registration).

Figura 4.16.4. Registro de nuevo usuario (web)

4.10.3 Registro Pacientes

El registro de los pacientes se puede observar en la figura 4.16.5, el cual se debe de realizar una vez que se ingresa en el sistema por parte de un Médico/enfermero, en este formulario se solicita la información básica del paciente, sus datos personales y algún medio de contacto, también aquí se asigna el número de placa que se encargará de monitorear su estado.

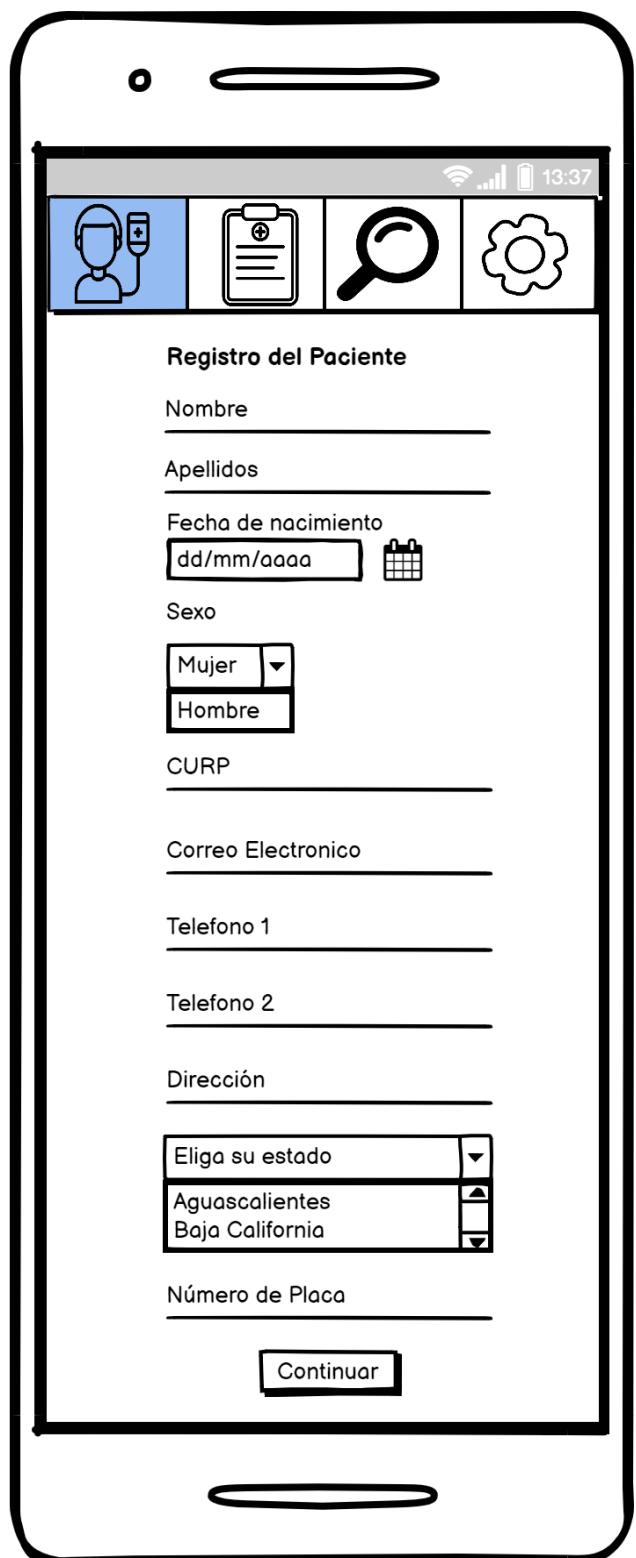


Figura 4.16.5. Registro de pacientes (móvil)

La figura 4.16.6 corresponde a la continuación del registro del paciente, esta información podrá decidir el paciente si darla o no, estos datos son para poder hacer el análisis de los datos para este paciente en particular. Tanto la figura 4.16.5 como la 4.16.6 corresponden a las versiones móviles del formulario para el registro del paciente.

Actividad Física

Actividad Física ▾
1 - Nula
5 - Mucha

Adicciones

Alcoholismo ▾
Tabaquismo

Antecedentes familiares

Diabetes ▾
Hipertensión

Consumo farmacos

Si ▾
No

Educación

Primaria ▾
Secunda

Estado civil

Soltero ▾
Casado

Exposición ruido

Poca ▾
Mucha

Exposición solar

Poca ▾
Mucha

Horas de sueño

2-3 horas ▾
4-5 horas
6-7 horas
8+ horas

Padecimientos

Diabetes ▾
Hipertensión

Personas dependientes

8 ▾
9
10+

Trabajo

Obrero ▾
Futbolista

Variaciones de Humedad

Poca ▾
Mucha

Variaciones de temperatura

Poca ▾
Mucha

Guardar

Figura 4.16.6. Registro de pacientes 2 (móvil)

La figura 4.16.7 es la versión web del registro del paciente, en primer lugar, se solicitará información básica del paciente.

The screenshot shows a web browser window with a light gray header bar containing standard navigation icons (back, forward, search, etc.). Below the header is a horizontal toolbar with four icons: a person icon, a clipboard icon, a magnifying glass icon, and a gear icon. The main content area is titled "Registro del Paciente". The form consists of several input fields and dropdown menus:

- Nombre:** Text input field.
- Apellidos:** Text input field.
- Fecha de nacimiento:** Text input field with placeholder "dd/mm/aaaa" followed by a calendar icon.
- Sexo:** A dropdown menu with two options: "Mujer" (selected) and "Hombre".
- CURP:** Text input field.
- Correo Electronico:** Text input field.
- Teléfono 1:** Text input field.
- Teléfono 2:** Text input field.
- Dirección:** Text input field.
- Eliga su estado:** A dropdown menu showing "Aguascalientes" and "Baja California" as options.

At the bottom center of the form is a "Continuar" (Continue) button.

Figura 4.16.7. Registro de pacientes (web)

La figura 4.16.8 es la versión web del formulario a llenar para la información del paciente, el cual será opcional y recibirá información algo más detallada sobre actividades que realiza el paciente, para poder hacer un mejor análisis de los datos.

The screenshot shows a web-based application for patient registration. At the top, there is a header bar with icons for back, forward, search, and settings. Below the header, the main form is divided into several sections:

- Actividad Física**: A dropdown menu showing "Actividad Física" with options "1 - Nula" and "5 - Mucha".
- Adicciones**: A dropdown menu showing "Alcoholismo" and "Tabaquismo".
- Antecedentes familiares**: A dropdown menu showing "Diabetes" and "Hipertensión".
- Consumo farmacos**: A dropdown menu showing "Si" and "No".
- Educación**: A dropdown menu showing "Primaria" and "Secundaria".
- Estado civil**: A dropdown menu showing "Soltero" and "Casado".
- Exposición ruido**: A dropdown menu showing "Poca" and "Mucho".
- Exposición solar**: A dropdown menu showing "Poca" and "Mucho".
- Horas de sueño**: A dropdown menu showing "2-3 horas", "4-5 horas", "6-7 horas", and "8+ horas".
- Padecimientos**: A dropdown menu showing "Diabetes" and "Hipertensión".
- Personas dependientes**: A dropdown menu showing "8", "9", and "10+".
- Trabajo**: A dropdown menu showing "Obrero" and "Futbolista".
- Variaciones de Humedad**: A dropdown menu showing "Poca" and "Mucho".
- Variaciones de temperatura**: A dropdown menu showing "Poca" and "Mucho".

At the bottom of the form is a button labeled "Guardar".

Figura 4.16.8. Registro de pacientes 2 (web)

4.10.4 Home (Lista de pacientes)

En la figura 4.16.9, se tiene una tabla que muestra una lista de los pacientes de algún Médico/enfermero, así como aquellas opciones que tiene sobre cada paciente, desde la visualización de su información, edición de esta misma o bien el despliegue de los signos vitales leídos en ese momento, también se muestra la ID del paciente y su nombre.

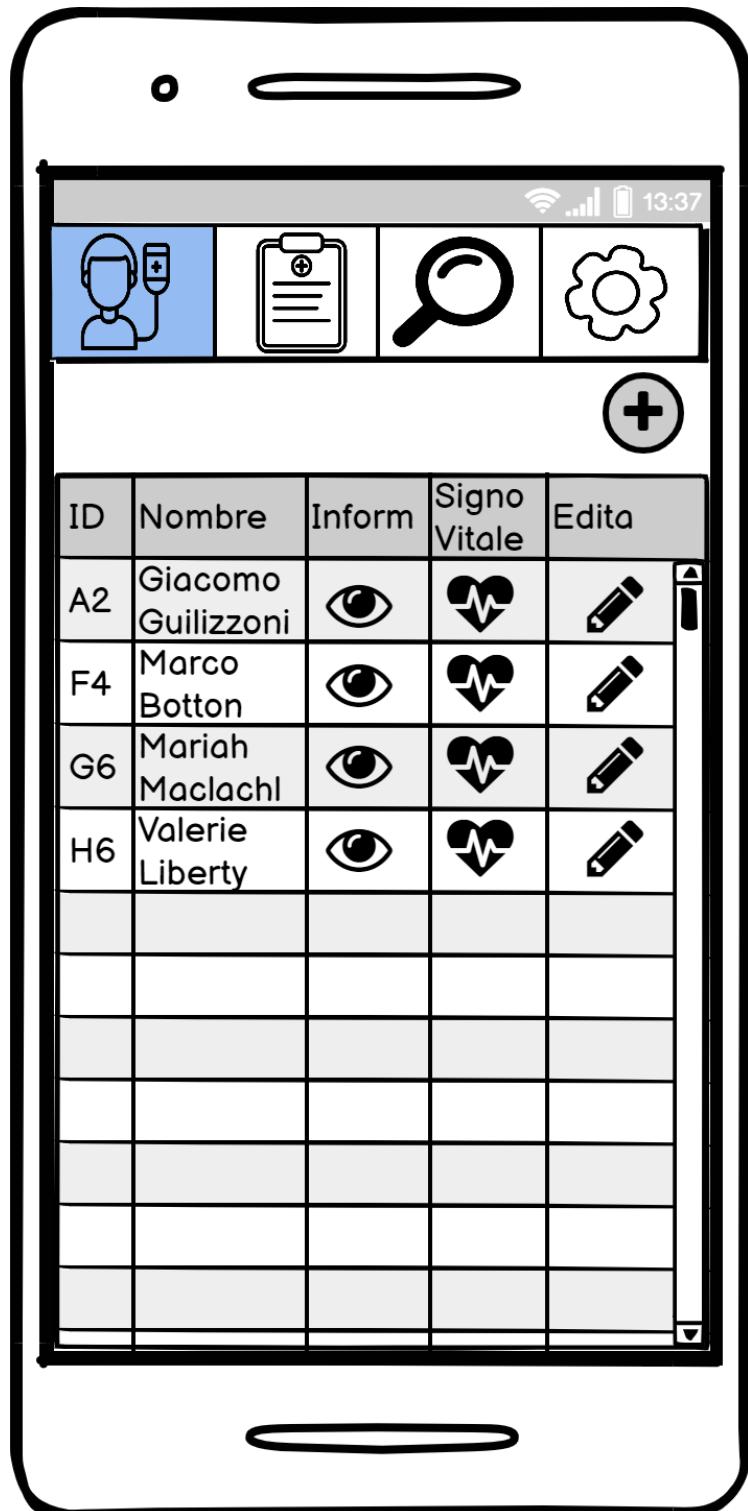


Figura 4.16.9. Lista de Pacientes (móvil)

A continuación, se muestra el mockup que corresponde a la lista de los pacientes, pero en este caso para la versión web, la figura 4.16.10 muestra esta tabla.

The mockup shows a web browser window with a header bar containing icons for back, forward, search, and refresh, and a 'Home' button. Below the header is a navigation bar with icons for user profile, clipboard, magnifying glass, and a flower-like icon. In the top right corner is a circular button with a plus sign. The main content area displays a table with the following data:

ID	Nombre	Informe	Signos Vitales	Editar
A22	Giacomo Guilizzoni	👁	❤️	✎
F44	Marco Botton	👁	❤️	✎
G66	Mariah MacLachlan	👁	❤️	✎
H66	Valerie Liberty	👁	❤️	✎

Figura 4.16.10. Lista de Pacientes (web)

4.10.5 Signos Vitales en tiempo real

Los signos vitales medidos en tiempo real serán mostrados en la interfaz gráfica del mockup de la figura 4.16.11. Este es uno de los mockups más importantes del sistema, ya que es una de las principales funcionalidades de Doc-Checker, se mostrarán los valores medidos en un color que se pueda familiarizar a un estado bueno o malo de los signos vitales, por ejemplo, color verde para valores dentro de un rango normal, rojo para valores dentro de un rango de gravedad, etc. Esta interfaz gráfica también muestra el ID del paciente y su nombre, el tamaño de los valores medidos también es un texto más grande para que sea más fácil de identificar el valor de cada signo vital, la versión de esta figura corresponde a la versión móvil.

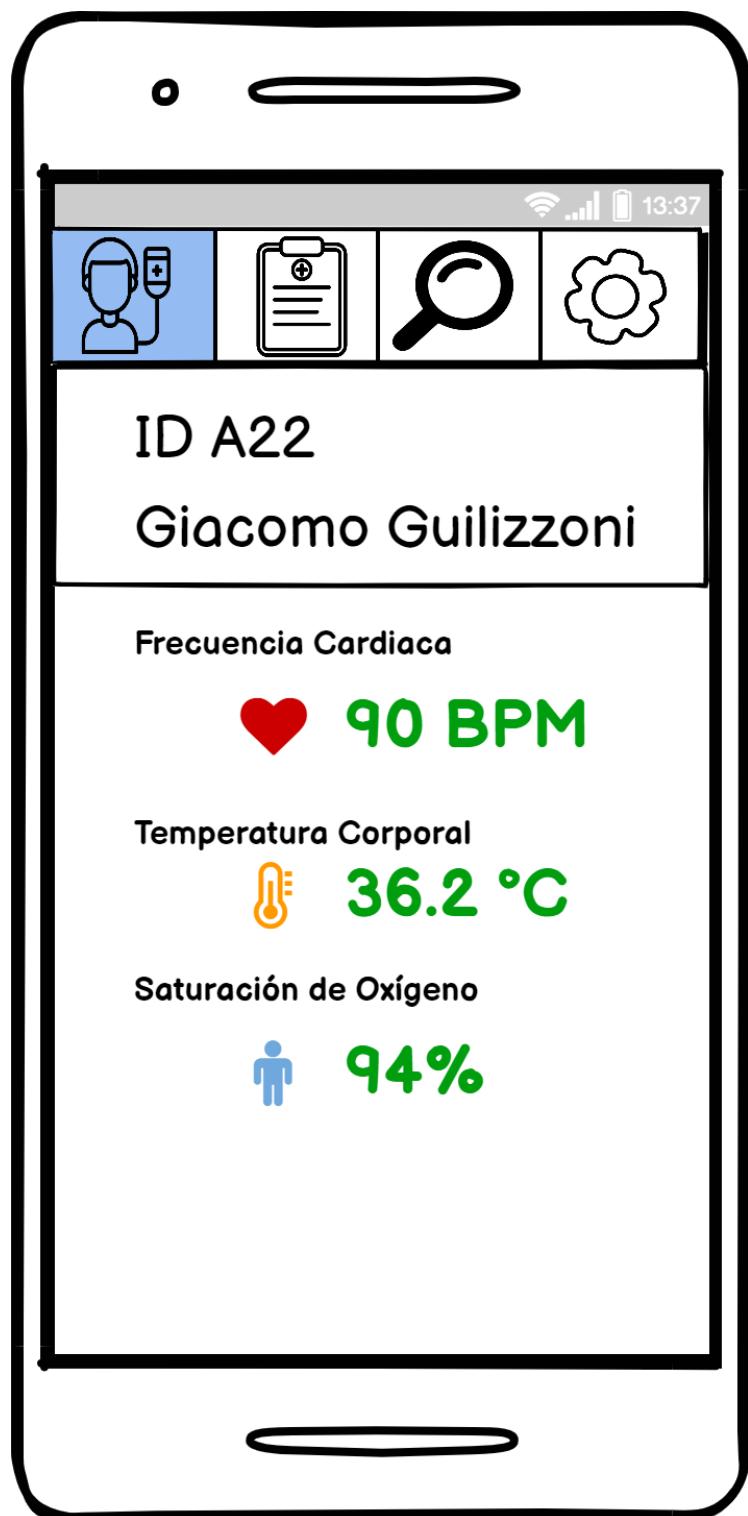


Figura 4.16.11. Signos Vitales (móvil)

La figura 4.16.12 corresponde a la versión web de la lectura de los signos vitales, en donde observamos una interfaz gráfica que aprovecha más el espacio para la versión web que la versión móvil.

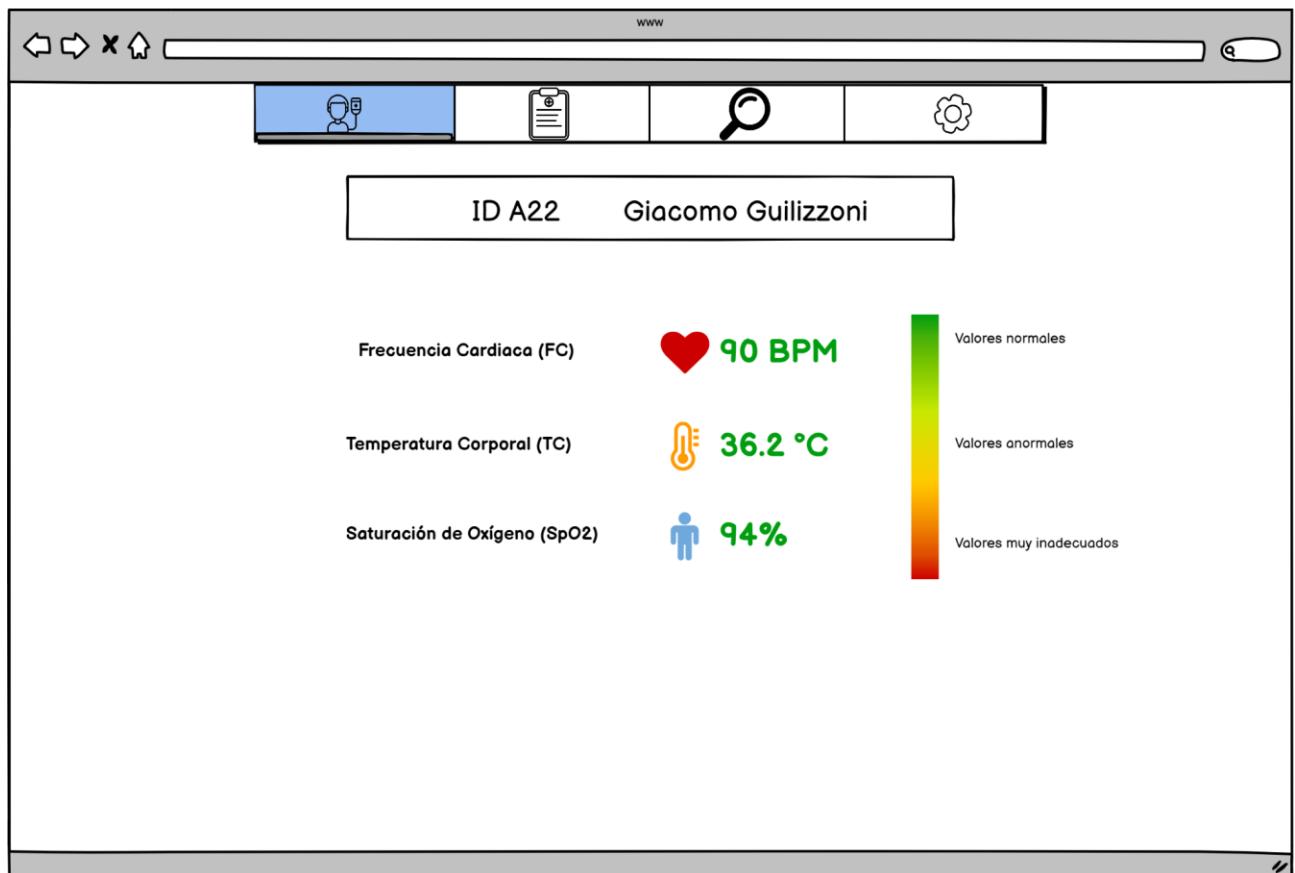


Figura 4.16.12. Signos Vitales (web)

4.10.6 Informe de paciente

En el mockup de la figura 4.16.13 se aprecia la interfaz gráfica correspondiente al despliegue de la información del paciente dentro del sistema para la versión móvil.



Figura 4.16.13. Informe de Paciente (móvil)

La figura 4.16.14 corresponde a la versión web del despliegue de la información del paciente dentro del sistema.

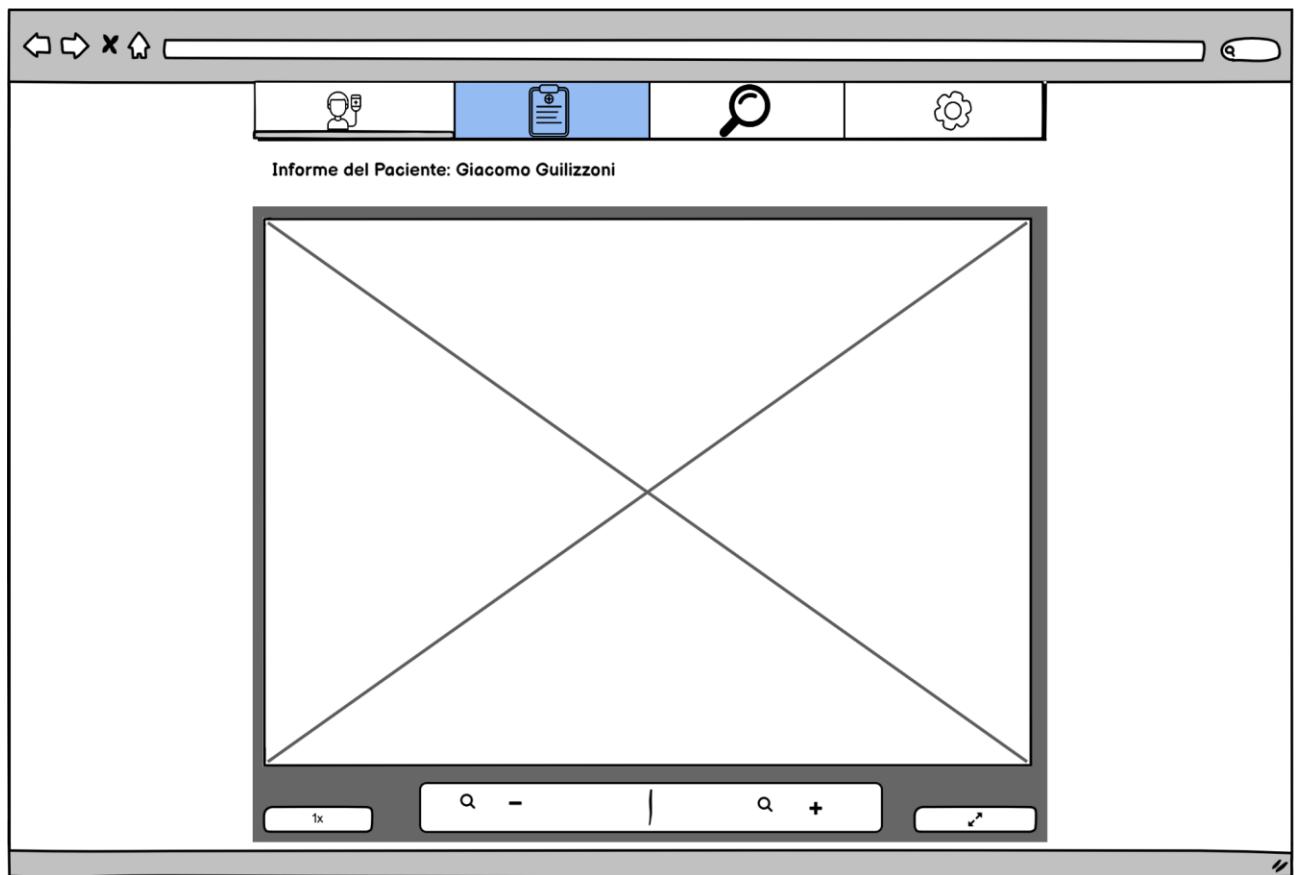


Figura 4.16.14. Informe de Paciente (web)

4.10.7 Configuración

En la figura 4.16.15 se muestra la pestaña de configuración para la aplicación móvil, en ella se podrán realizar algunas tareas como modificación de la contraseña de acceso al perfil, información de la aplicación y el botón para cerrar la sesión, para poder acceder desde otro perfil.

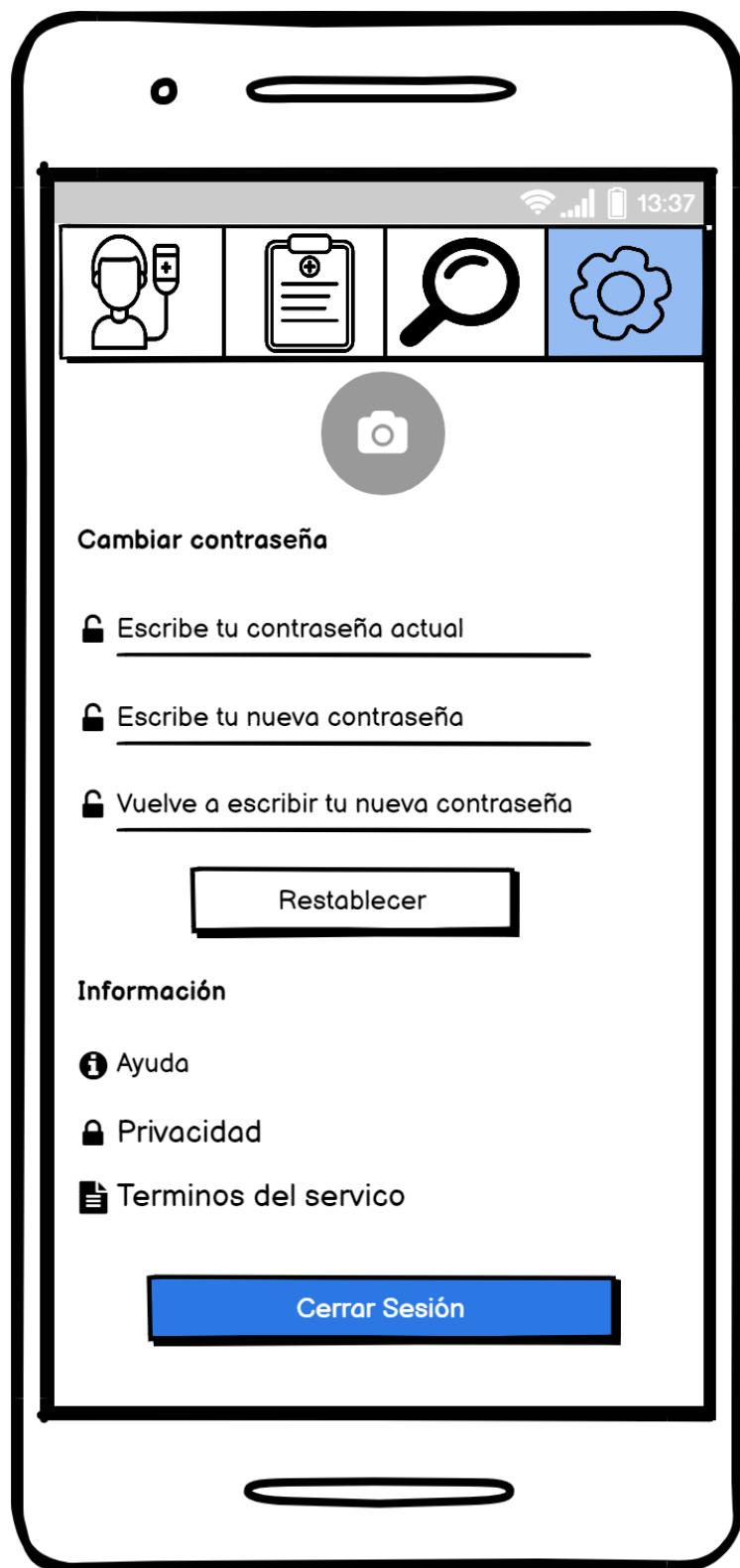


Figura 4.16.15. Configuración (móvil)

La figura 4.16.16 muestra la versión web de la pestaña para la configuración de la aplicación.

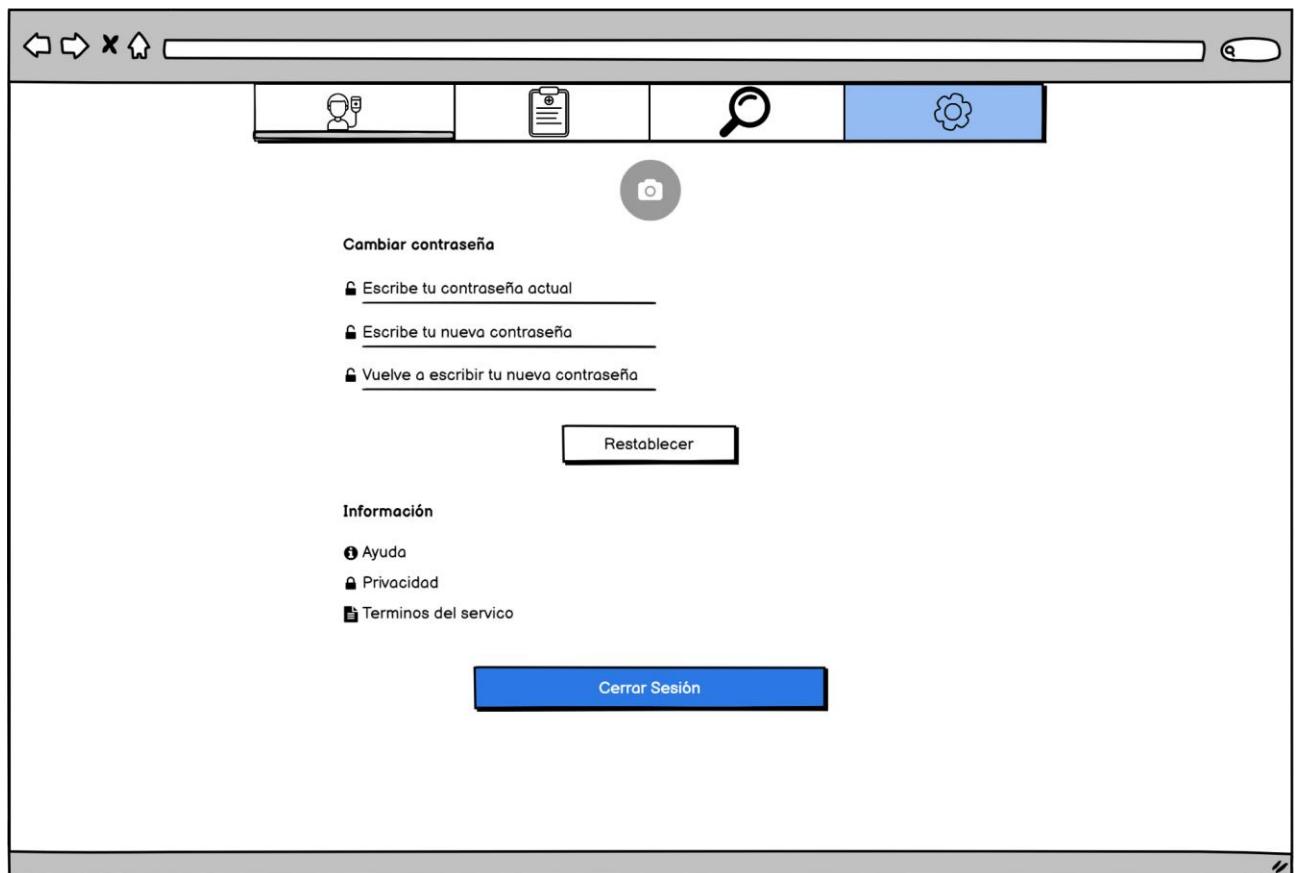


Figura 4.16.16. Configuración (web)

4.10.8 Consulta de Reporte de Minería de Datos

Una de las tareas más importantes para este proyecto es la posibilidad de realizar un análisis sobre los datos recabados de los pacientes, lo cual se podrá mostrar en un archivo PDF que contenga esta información que previamente fue analizada, en la figura 4.16.17 se muestra la versión móvil para el despliegue del reporte de minería de datos.

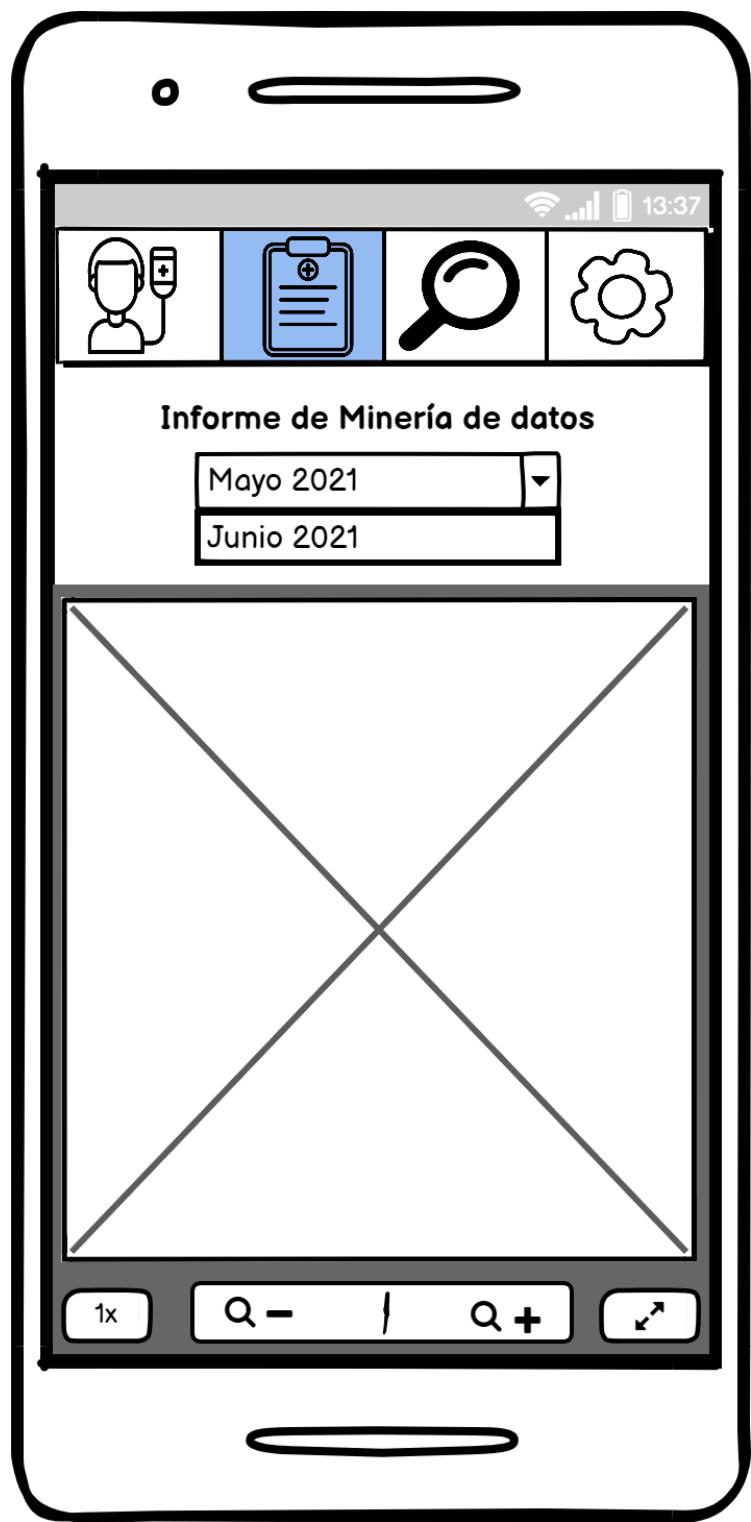


Figura 4.16.17. Reporte de Minería de Datos (móvil)

En la figura 4.16.18 se muestra la versión web para el despliegue del reporte de minería de datos.

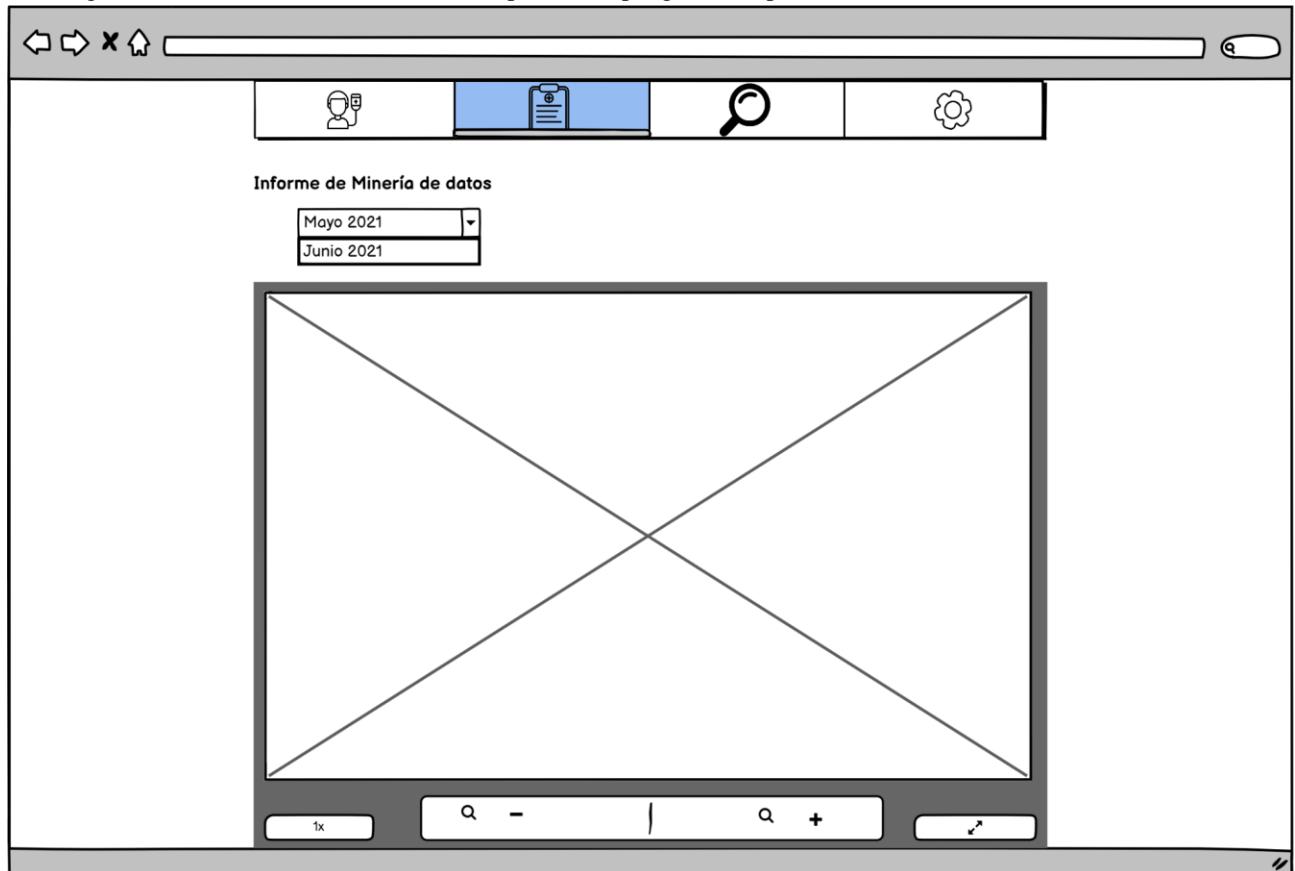


Figura 4.16.18. Reporte de Minería de Datos (web)

4.10.9 Búsqueda de pacientes

En el mockup de la figura 4.16.19 se observa el buscador del sistema, en el que se escribirá la id del paciente y podrá ser agregado a la lista de pacientes del Médico(a) o Enfermero(a) de ese usuario, corresponde a la versión móvil y se desplegará una lista de pacientes que coincidan con la búsqueda y que el Médico/enfermero pueda consultar, los pacientes que podrán consultar serán aquellos que se encuentren en el mismo hospital en el que ellos trabajen.

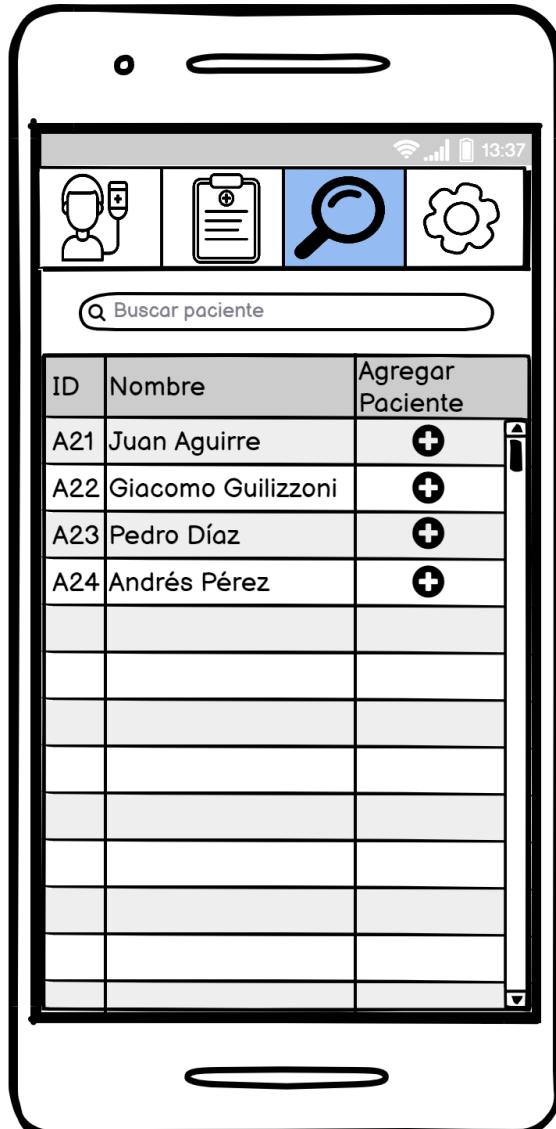


Figura 4.16.19. Búsqueda de Pacientes (móvil)

La figura 4.16.20 es la versión web del buscador de pacientes, se observa un aumento en el tamaño de la tabla, en la que se puede visualizar de una mejor forma el nombre de cada paciente.

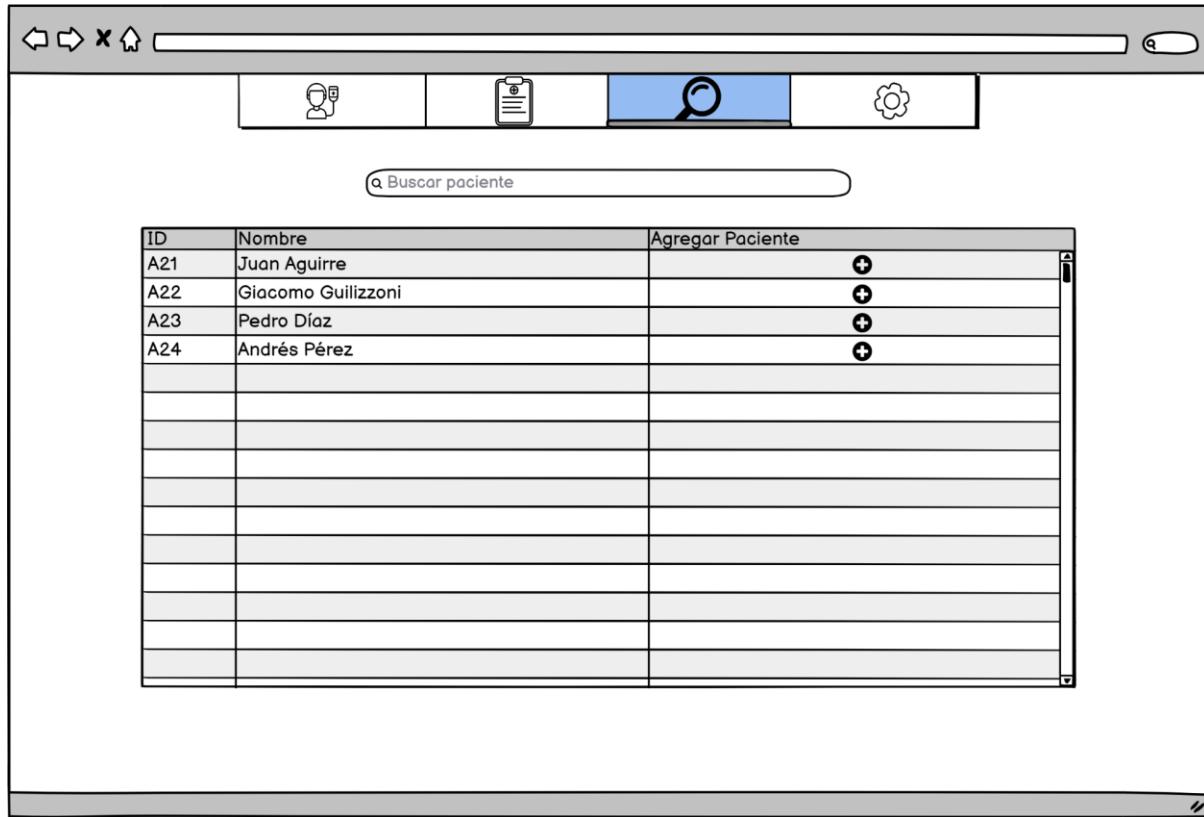


Figura 4.16.20. Búsqueda de Pacientes (web)

4.10.10 Pestaña administrador

La figura 4.16.21 es la pestaña del administrador que solamente estará disponible en la versión web, en la que tendrá dos botones indicando las solicitudes de los usuarios que se registraron previamente y el otro botón para registrar placas nuevas.

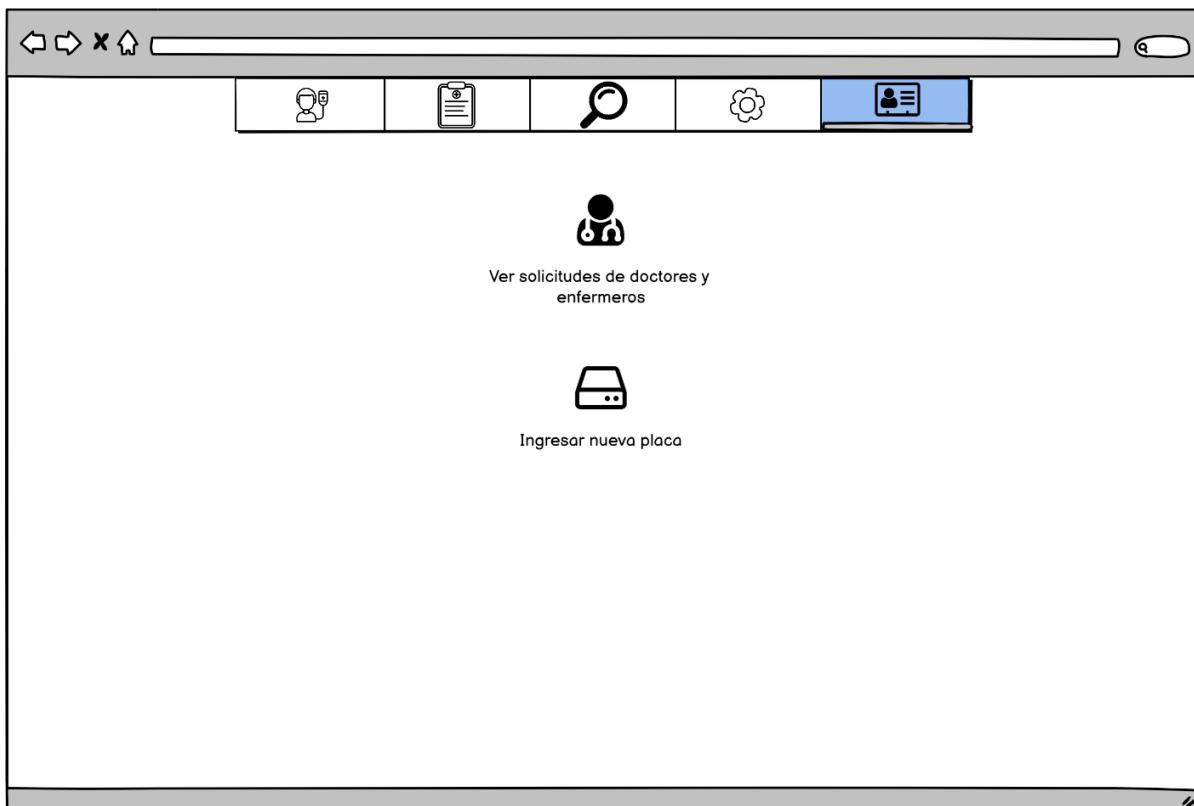


Figura 4.16.21. Pestaña Administrador (web)

4.10.11 Solicitud de Médicos

La figura 4.16.22 muestra las solicitudes de los usuarios registrados donde el administrador es el encargado de aceptar o rechazar cada solicitud.

Figura 4.16.22. Solicitudes de Médicos (web)

4.10.12 Ingreso de placas

La figura 4.16.23 muestra el formulario en el que se ingresarán las placas y en el hospital en donde se encuentra.

← → × ↻



Registro de la placa

ID de la placa

Hospital

Guardar

Figura 4.16.23. Ingreso de Placas (web)

4.10.13 Búsqueda de pacientes en administrador

La figura 4.16.24 muestra las pestañas de búsqueda de pacientes del administrador en la que mostrarán a todos los pacientes y los podrá eliminar.

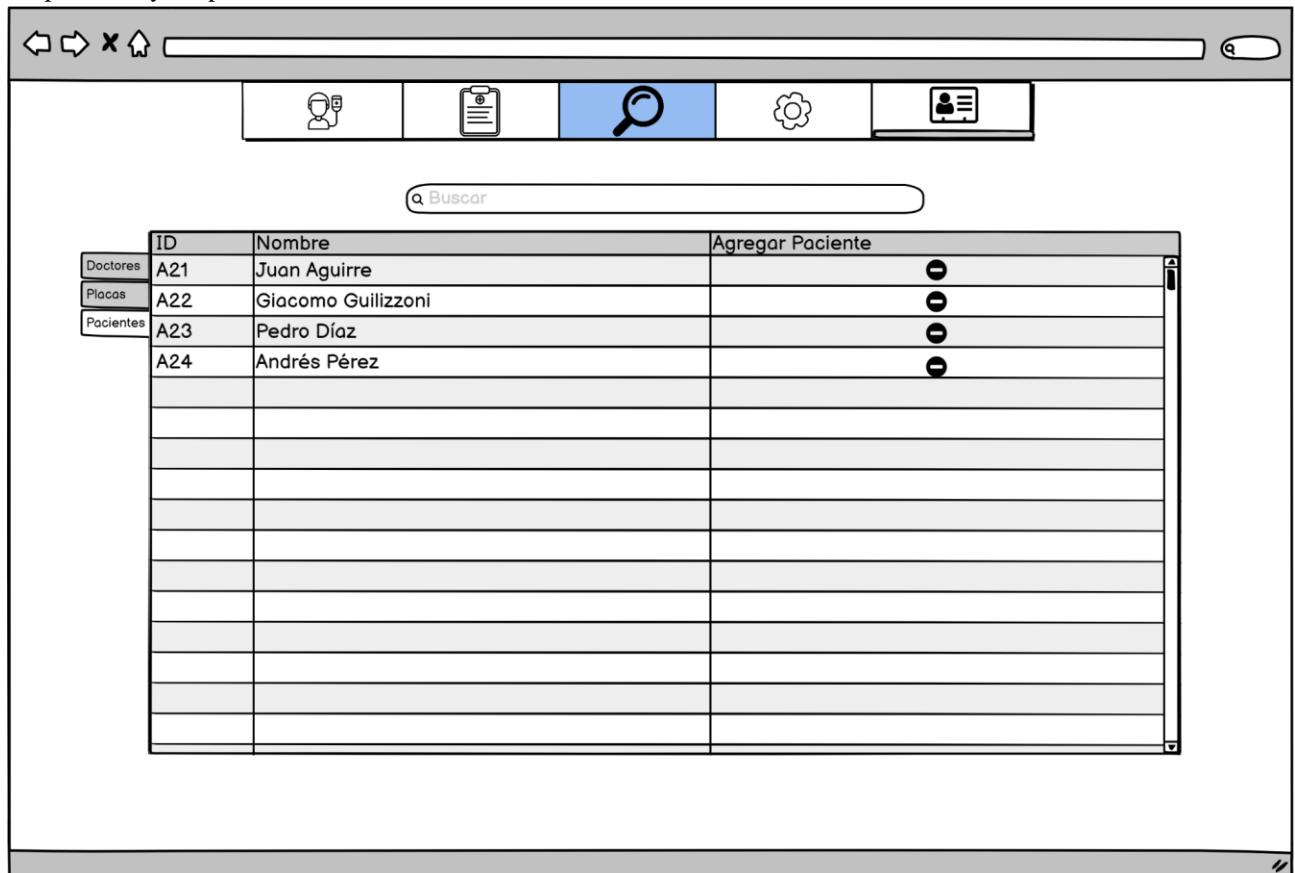


Figura 4.16.24. Búsqueda de Pacientes (web)

4.10.14 Búsqueda de Médicos en administrador

La figura 4.16.25 muestra las pestañas de búsqueda de Médicos del administrador en la que mostrarán a todos los Médicos, con su ID del sistema, nombre completo, cédula profesional y se podrán eliminar.

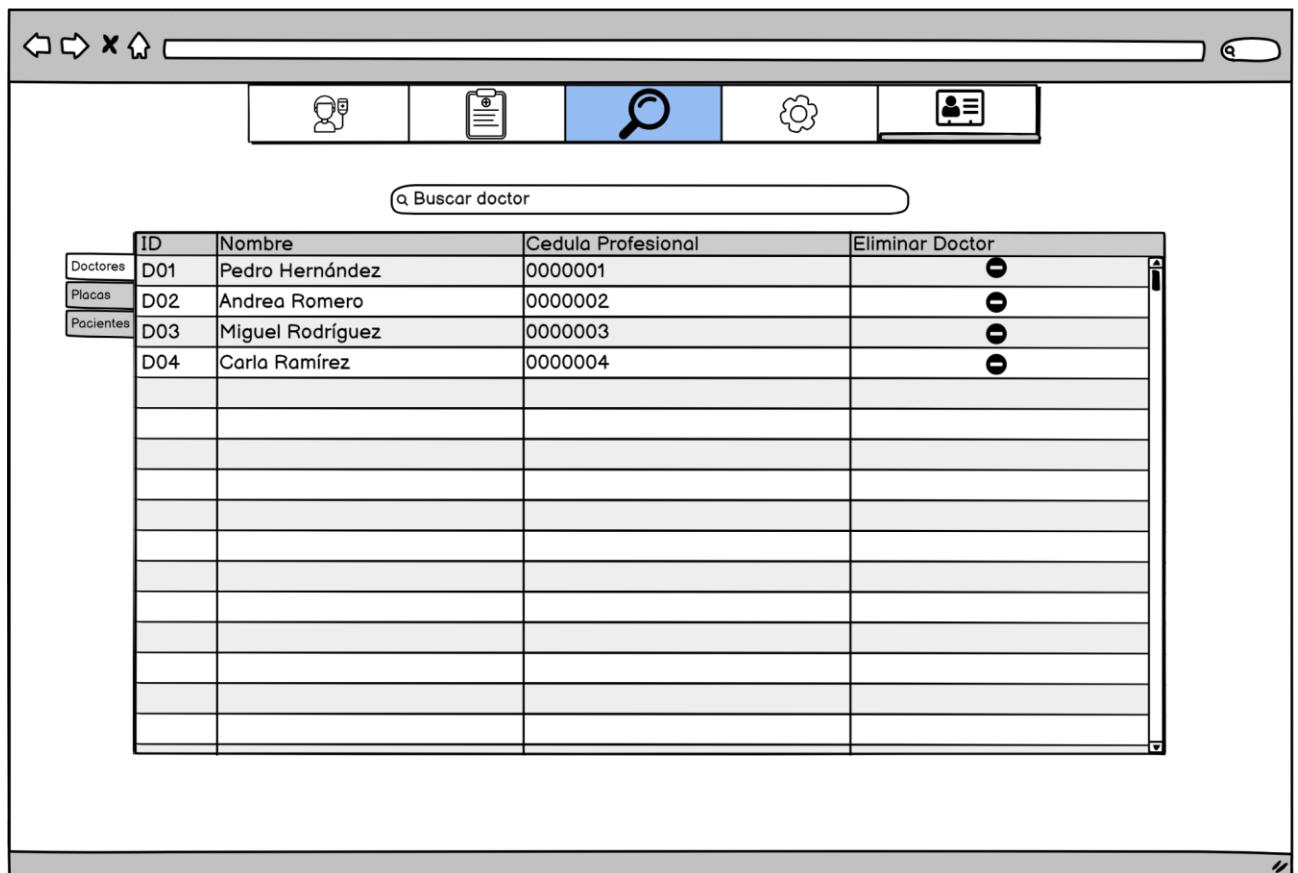


Figura 4.16.25. Búsqueda de Médicos (web)

4.10.15 Búsqueda de placas en administrador

La figura 4.16.26 presenta las pestañas de búsqueda de placas del administrador en la que se ilustrarán todas las placas y el nombre del hospital en donde se encuentran.

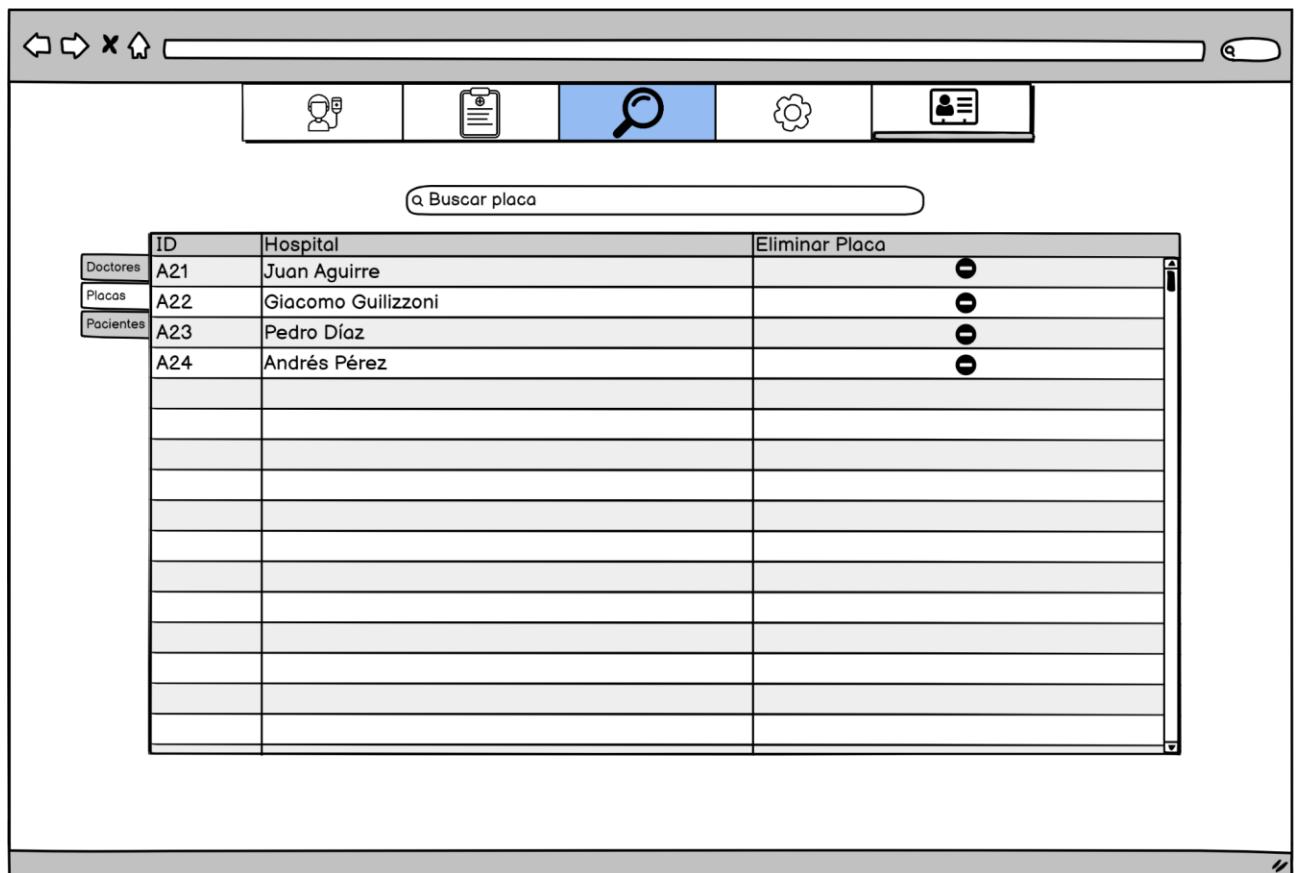


Figura 4.16.26. Búsqueda de placas en administrador (web)

4.11 Diagrama de flujo del algoritmo a programar en la placa de desarrollo

La figura 4.17 muestra el algoritmo que se programará en la placa de desarrollo, se puede observar que cuando comienza el funcionamiento del algoritmo se leen los valores de los sensores y después se hace una validación para saber si los valores que se están leyendo se encuentran en un rango adecuado para saber si informar al servidor o bien continuar con su funcionamiento ordinario. Dentro de su funcionamiento ordinario la placa acepta conexiones de nuevos clientes o envía los promedios de sus lecturas sobre los signos vitales a sus distintos clientes.

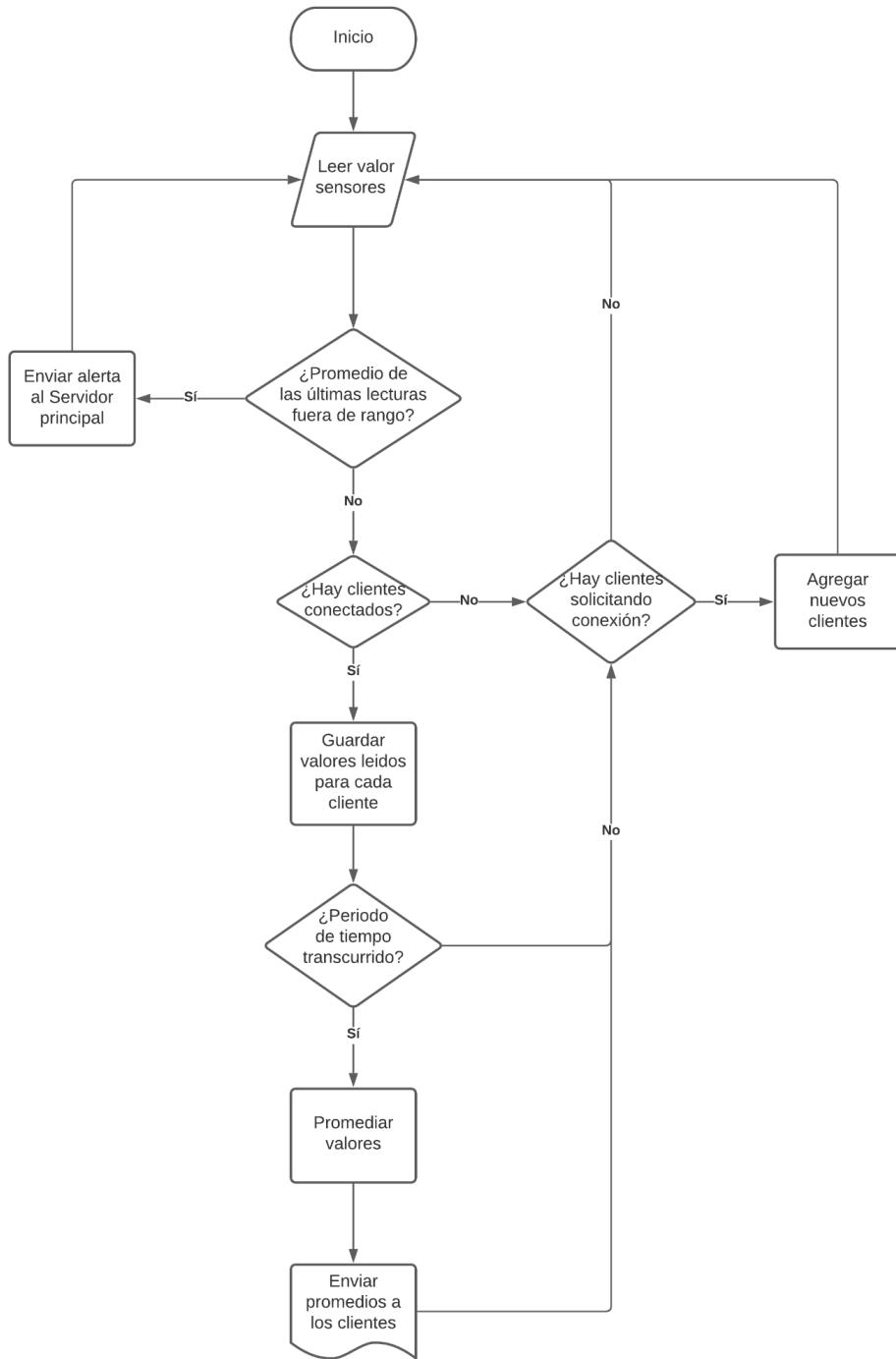


Figura 4.17. Algoritmo por programar en la placa de desarrollo

Capítulo V: Desarrollo del sistema

5.1 Módulo de sensado

Para la entrega del Trabajo Terminal 1 el equipo de desarrollo logró concretar la comunicación a través de I2C con los sensores y la placa de desarrollo, la figura 5.1 muestra la conexión física entre los sensores y la placa de desarrollo, se puede apreciar que la placa de desarrollo suministra la alimentación para los sensores y ambos sensores se conectan a dos pines de la placa, estos son el D21 y D22, estos pines incorporan el protocolo de comunicación I2C.

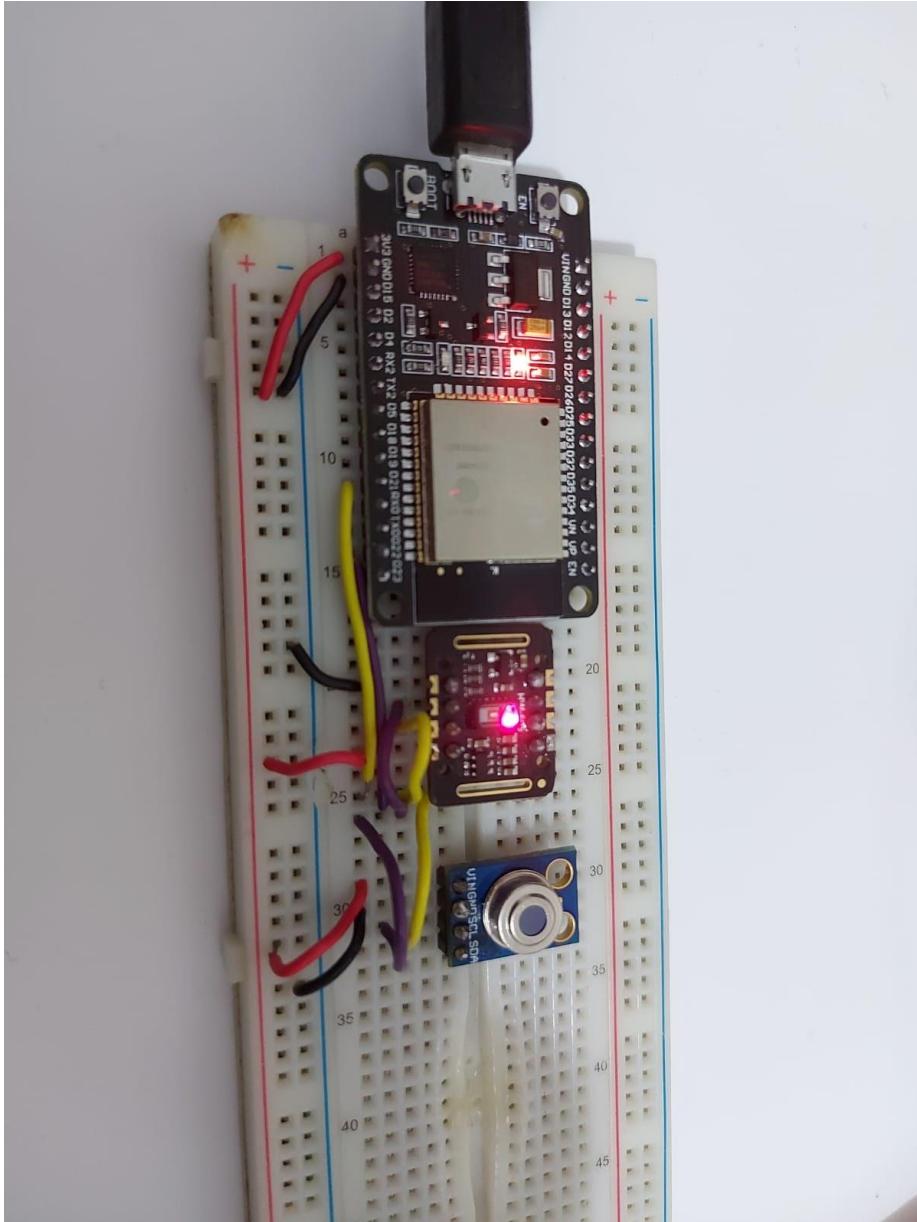
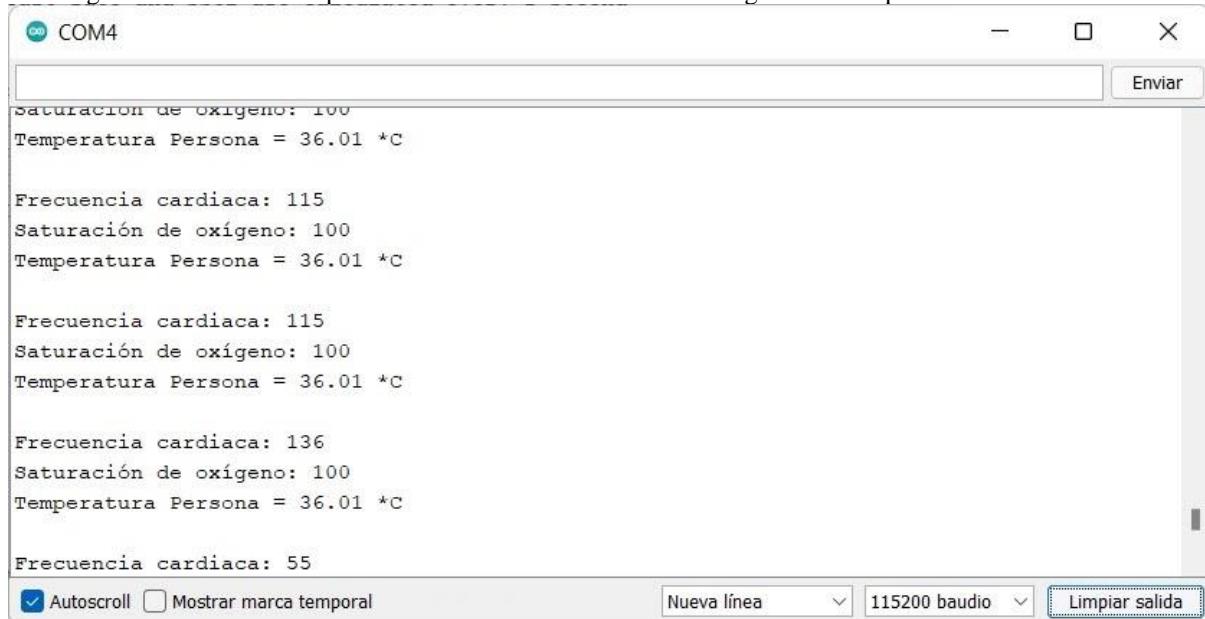


Figura 5.1. Placa de desarrollo y conexión física con los sensores

En la figura 5.2 se muestra la primera versión de la lectura de los 3 signos vitales que serán enviados al servidor



The screenshot shows a terminal window titled "COM4". The window displays the following text output:

```
Saturación de oxígeno: 100
Temperatura Persona = 36.01 *C

Frecuencia cardiaca: 115
Saturación de oxígeno: 100
Temperatura Persona = 36.01 *C

Frecuencia cardiaca: 115
Saturación de oxígeno: 100
Temperatura Persona = 36.01 *C

Frecuencia cardiaca: 136
Saturación de oxígeno: 100
Temperatura Persona = 36.01 *C

Frecuencia cardiaca: 55
```

At the bottom of the window, there are several control buttons: "Autoscroll" (checked), "Mostrar marca temporal" (unchecked), "Nueva línea" (dropdown menu), "115200 baudio" (dropdown menu), and "Limpiar salida" (button).

Figura 5.2. Salida por el puerto serial de la placa

5.2 Aplicación híbrida

En el proceso de realización de la aplicación híbrida, se desarrolló la interfaz gráfica que corresponde al inicio de sesión como se puede observar en la Figura 5.3 la cual corresponde a la interfaz gráfica web y la Figura 5.4 que corresponde a la interfaz gráfica móvil.

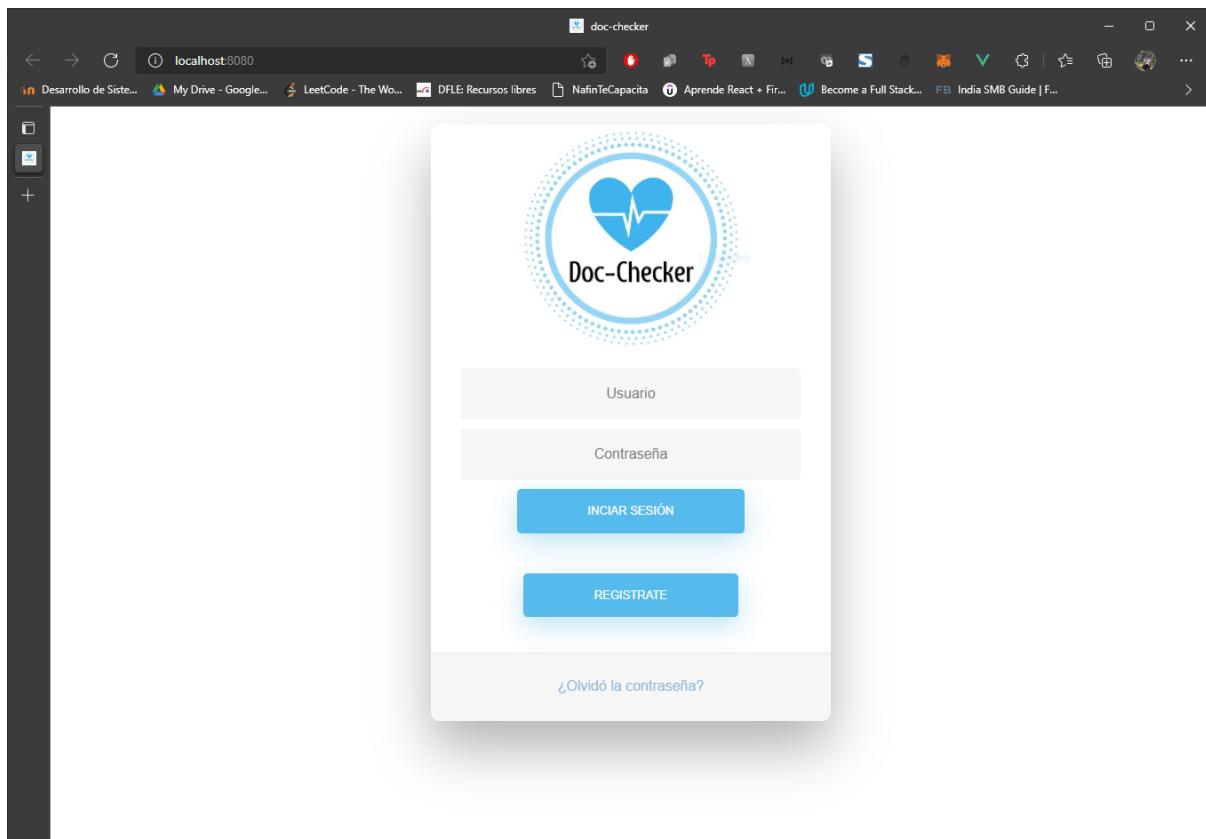


Figura 5.3 Versión web de la Aplicación híbrida

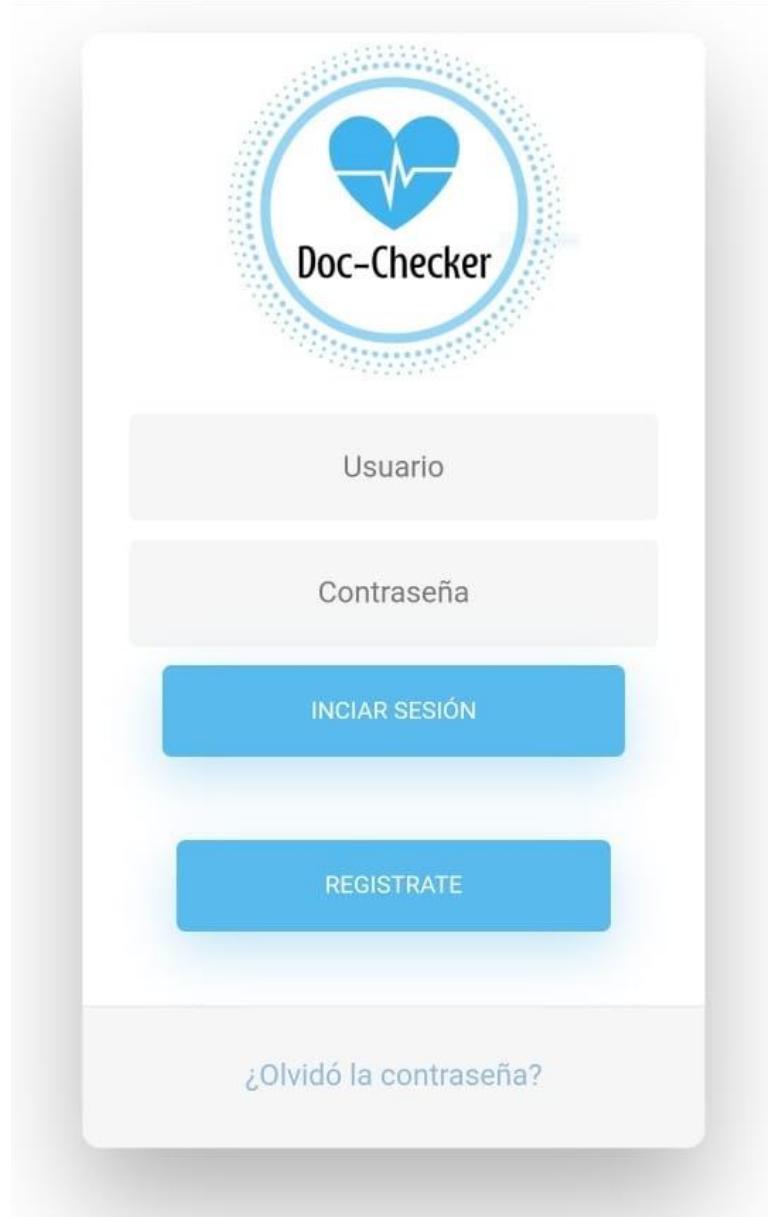


Figura 5.4 Versión Móvil de la Aplicación híbrida

Referencias

- [1] M. Fernández Jaimes, R. Zárate Grajales, J. Ochoa Cervantes and M. Ramírez Antonio, "La evaluación de la calidad de los signos vitales como indicador de proceso en la Gestión del Cuidado de Enfermería", Revista Mexicana, CDMX, 2010.
- [2] N. Cruz Serrano, "En el IMSS, equipo obsoleto pone en riesgo a pacientes", *El Universal*, 2021. [En línea]. Disponible en: <https://www.eluniversal.com.mx/cartera/en-el-imss-equipo-obsoleto-pone-en-riesgo-pacientes>. [Accesado: 29- May- 2021].
- [3] Instituto Nacional de Migración, "Temas de 1a Plana 241219", inm.gob.mx, 2019. [En Línea]. Disponible en: <https://www.inm.gob.mx/gobmx/word/index.php/temas-de-1a-plana-241219/>. [Accesado: 29- May- 2021].
- [4] S. Aust, "Best practice: evidence based practice information sheets for health professionals." Joanna Briggs Institute for Evidence Based Nursing, 2009. Disponible en: https://www.evidenciaencuidados.es/BPIS/PDF/1999_3_3_SignosVitales.pdf [Accesado: 20-Ago-2021]
- [5] R. Pressman, "Ingeniería del software: Un enfoque práctico", New York: McGraw-Hill, 2010, 7th ed., pp. 35.
- [6] Pérez, A et al, "Una metodología para el desarrollo de hardware y software embebidos en sistemas críticos de seguridad", Systemics, Cybernetics and Informatics Journal, vol. 3, Núm. 2, 2006, pp. 70-75.
- [7] "PM-7000D Detalles del producto", Zoncare.com, 2021. [En línea]. Disponible en: http://www.zoncare.com/html/sp/cp/xd/jhy/PM_7000D/. [Accesado: 18- May- 2021].
- [8] Azumio Inc., "Ritmo Cardíaco - Monitor Pulso". Google Play. [En línea]. Disponible en: https://play.google.com/store/apps/details?id=si.modula.android.instantheartrate&hl=es_MX&gl=US. [Accesado: 17-Mayo-2021].
- [9] H. García et al., "Monitor de signos vitales con comunicación inalámbrica Wi-Fi para unidad de cuidados intensivos desarrollado en LabVIEW y la tarjeta myRIO-1900", CIC, 2018. [En línea]. Disponible en: https://rcs.cic.ipn.mx/2018_147_12/Monitor%20de%20signos%20vitales%20con%20comunicacion%20inalambrica%20Wi-Fi%20para%20unidad%20de%20cuidados.pdf. [Accesado: 17-Mayo-2021].
- [10] García, J. (2021, 4 de mayo). Xiaomi Mi Watch, análisis: un claro aspirante al podio como mejor smartwatch en calidad precio. Xataka - Tecnología y gadgets, móviles, informática, electrónica. Disponible en: <https://www.xataka.com/analisis/xiaomi-mi-watch-analisis-caracteristicas-precio-especificaciones> [Accesado: 17-Oct-2021]
- [11] García, J. (2021b, 3 de septiembre). Samsung Galaxy Watch 4, análisis: la vuelta a Wear OS le sienta de escándalo al nuevo smartwatch de Samsung. Xataka - Tecnología y gadgets, móviles, informática, electrónica. Disponible en: <https://www.xataka.com/analisis/samsung-galaxy-watch-4-analisis-caracteristicas-precio-especifcaciones> [Accesado: 17-Oct-2021]
- [12] Martí, A. (2021, 27 de mayo). Apple Watch Series 6, análisis: el Apple Watch de siempre para usuarios exigentes como nunca. Xataka - Tecnología y gadgets, móviles, informática, electrónica. Disponible en: <https://www.xataka.com/analisis/apple-watch-series-6-analisis-caracteristicas-precio-especificaciones> [Accesado: 17-Oct-2021]
- [13] Villegas González, Juliana, Villegas Arenas, Oscar Alberto, Villegas González, Valentina Semiología de los signos vitales: Una mirada novedosa a un problema vigente. Archivos de Medicina (Col) [en línea]. 2012, 12(2), 221-240. ISSN: 1657-320X. Disponible en: <https://www.redalyc.org/articulo.oa?id=273825390009> [Accesado: 9-Oct-2021]
- [14] Santiago, A. (2020, 12 de febrero). "Control de temperatura por medios físicos". yoamoenfermeria.com. [En línea]. Disponible en: <https://yoamoenfermeriablog.com/2020/02/12/control-de-temperatura/> [Accesado: 3-Sep-2021]
- [15] Talamas, J. (s. f.). "Práctica #3.1: TOMA DE SIGNOS VITALES". Facultad de Medicina y Nutrición UJED. [En línea]. Disponible en: http://famen.ujed.mx/doc/manual-de-practicas/a-2016/03_Prac_01.pdf [Accesado: 8-Sep-2021]
- [16] Ministerio de salud. (2020, 12 de octubre). "DOCUMENTO TÉCNICO: MANEJO DE PERSONAS AFECTADAS POR COVID-19 EN LOS SERVICIOS DE HOSPITALIZACIÓN". Portal Regional da BVS. [En línea]. Disponible en: <https://docs.bvsalud.org/biblioref/2020/10/1122354/rm-839-2020.pdf> [Accesado:10-Sep-2021]

- [17] "Centro Nacional de Excelencia Tecnológica en Salud". CENETEC. Disponible en: http://www.cenetec.salud.gob.mx/descargas/biomedica/guias_tecnologicas/13gt_monitores.pdf [Accesado:11-sep-2021].
- [18] "¿Que es un servidor y para qué sirve?" Informática para tu negocio. Disponible en: <https://www.informaticaparatunegocio.com/blog/que-es-un-servidor-y-para-que-sirve/> [Accesado:11-sep-2021].
- [19] "¿Qué es un socket?" Speed Test by Speedcheck - Test your internet speed. Disponible en: <https://www.speedcheck.org/es/wiki/socket/> ([Accesado: 12-Sep-2021].
- [20] Oracle. "¿Qué es Internet of Things (IoT)?" Oracle | Integrated Cloud Applications and Platform Services. [En línea]. Disponible en: <https://www.oracle.com/mx/internet-of-things/what-is-iot/#link0> [Accesado: 12-Sep-2021].
- [21] R. Kamal, Embedded Systems, 2nd ed. New York: Tata McGraw-Hill, 2008, pp. 4-6.
- [22] "Aplicación híbrida o app híbrida". Computer Weekly. Disponible en: <https://www.computerweekly.com/es/definicion/Aplicacion-hibrida-o-app-hibrida> [Accesado: 12-Sep-2021].
- [23] ¿Qué es una base de datos? (2021). Oracle | Integrated Cloud Applications and Platform Services. <https://www.oracle.com/mx/database/what-is-database/> [Accesado: 12-Sep-2021]
- [24] "Datamining (Minería de datos)". Sinnexus. Disponible en: https://www.sinnexus.com/business_intelligence/datamining.aspx [Accesado: 12-Sep-2021].
- [25] "LA MATRIZ DE GESTIÓN Y CONTROL DE RIESGOS: HERRAMIENTA EFECTIVA EN LA AUDITORÍA INTERNA PARA LAS EMPRESAS COMERCIALES DE LA CIUDAD DE PANAMÁ". Índice de revistas en consolidación AmeliCA. Disponible en: <http://portal.amelica.org/ameli/jatsRepo/221/2212240002/html/index.html> [Accesado: 12-Sep-2021].
- [26] Adriana Gómez, María del C. López, Silvina Migani, Alejandra Otazú. "UN MODELO DE ESTIMACIÓN DE PROYECTOS DE SOFTWARE | Teoría." Disponible en: <https://blogadmi1.files.wordpress.com/2010/11/cocom0llfull.pdf> [Accesado: 12-Sep-2021].
- [27] F. Lizano-Madriz. "Estimación de costo de software: Una propuesta de aplicación pedagógica de COCOMO". Sistema de Información Científica Redalyc, Red de Revistas Científicas. Disponible en: <https://www.redalyc.org/journal/4759/475958170008/html/> [Accesado: 12-Sep-2021].
- [28] T. A. Antúnez Barbosa. "Scielo". Estimación de costos de desarrollo, caso de estudio: Sistema de Gestión de Calidad del Reactor TRIGA Mark III. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992016000100018 [Accesado: 12-Sep-2021].
- [29] Pradel Miquel, J. (s. f.). Ingeniería del software. Disponible en: http://cv.uoc.edu/annotation/68cdc4848e7fd4149910ab359b7a710d/699675/PID_00232008/PID_00232008.htm!#w30aac11b9b7c11 [Accesado: 12-Sep-2021]
- [30] Redactor Rock Conter. "¿Qué es Java y cuáles son sus características? - Rock Content". Rock Content - ES. [En línea]. Disponible en: <https://rockcontent.com/es/blog/que-es-java/> [Accesado: 5-Sep-2021].
- [31] R. Ramos. "¿Qué es JavaScript y para qué sirve?" Agencia de Marketing Digital Sevilla. [En línea]. Disponible en: <https://soyrafaramos.com/que-es-javascript-para-que-sirve/> [Accesado: 4-Sep-2021].
- [32] C. Herrera. "Ventajas y desventajas de usar JavaScript en la programación web". Blogueropro. [En línea]. Disponible en: <https://blogueropro.com/blog/ventajas-y-desventajas-de-usar-javascript-en-la-programacion-web> [Accesado: 4-Sep-2021].
- [33] K. Casey. "Comparación de 6 de los principales lenguajes de programación". ComputerWeekly.es. [En línea]. Disponible en: <https://searchdatacenter.techtarget.com/es/consejo/Comparacion-de-6-de-los-principales-lenguajes-de-programacion> [Accesado: 9-Sep-2021].
- [34] C. Valdeolmillos. "Las apps para Android desarrolladas en Python pueden ser realidad antes de lo que crees". MuyComputerPRO. [En línea]. Disponible en: <https://www.muycomputerpro.com/2020/05/12/apps-android-desarrolladas-python> [Accesado: 9-Sep-2021].
- [35] R. Hernandez. "Kivy con Python para el desarrollo de aplicaciones móviles - ▷ Cursos de Programación de 0 a Experto © Garantizados". ▷ Cursos de Programación de 0 a Experto © Garantizados. [En línea]. Disponible en: <https://unipython.com/kivy-python-desarrollo-aplicaciones-moviles/> [Accesado: 9-Sep-2021].
- [36] J. M. Agüero Atmitim. "Qué es Ionic: ventajas y desventajas de usarlo en apps móviles híbridas". Profile Software Services. Disponible en: <https://profile.es/blog/que-es-ionic/> [Accesado: 12-Sep-2021].

- [37] J. A. Blanes. "¿Qué es React Native?" Deloitte Spain. Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-react-native.html> [Accesado: 12-Sep-2021].
- [38] "React Native: Ventajas y desventajas de este framework". Rootstack | Web, Mobile & Automation solutions agency based in Panama with global clients | Rootstack. Disponible en: <https://www.rootstack.com/es/blog/react-native-ventajas-y-desventajas-de-este-framework/> [Accesado: 12-Sep-2021].
- [39] K. Estrada. "Por que es Vue.js el nuevo framework de moda". Medium. Disponible en: <https://medium.com/blog-apside/por-que-es-vue-js-es-el-nuevo-framework-de-moda-79de70e13ef5> [Accesado: 12-Sep-2021].
- [40] A. Pérez. "¿Por qué Desarrollar un Proyecto Front End en Angular?" DIGITAL55. Disponible en: <https://www.digital55.com/desarrollo-tecnologia/desarrollar-proyecto-front-end-angular/> [Accesado: 12-Sep-21]
- [41] "ReactJS vs Angular5 vs Vue.js ¿Cuál elegir? – Developing in Spanish". Developing in Spanish – Noticias, artículos, tutoriales y más sobre programación... en español. Disponible en: <http://developinginspanish.com/2019/03/08/reactjs-vs-angular5-vs-vue-js-cual-elegir/> [Accesado: 12-Sep-2021].
- [42]"Introducción a Android Studio | Desarrolladores de Android", Android Developers. [En línea]. Disponible en: <https://developer.android.com/studio/intro?hl=es-419>. [Accesado: 13- Sep- 2021].
- [43]"Visual Studio Code vs Android Studio: funcionalidad, búsqueda y control de código fuente", ICHI.PRO, 2019. [En línea]. Disponible en: <https://ichi.pro/es/visual-studio-code-vs-android-studio-funcionalidad-busqueda-y-control-de-codigo-fuente-121692821821526>. [Accesado: 13- Sep- 2021].
- [44] "Visual Studio Code - EcuRed", Ecured.cu, 2020. [En línea]. Disponible en: https://www.ecured.cu/Visual_Studio_Code. [Accesado: 13- Sep- 2021].
- [45] Velasco, R. (2021, 26 de mayo). "Visual Studio Code: el editor de código de Microsoft que querrás instalar". SoftZone. [En línea]. Disponible en: <https://www.softzone.es/programas/utilidades/visual-studio-code/> [Accesado: 13-Sep-2021]
- [46] Fenollosa, A. (2017, 4 de julio). "Vim vs Visual Studio Code". Programador Web Valencia. [En línea]. Disponible en: <https://programadorwebvalencia.com/vim-vs-vscode/> [Accesado: 16-Sep-2021]
- [47] "Sublime Text: Información y trucos para empezar desde cero", CIPSA Academia Cursos Informática en Barcelona y Bilbao, 2018. [En línea]. Disponible en: <https://cipsa.net/sublime-text-informacion-y-trucos-para-empezar-desde-cero/>. [Accesado: 16- Sep- 2021].
- [48] "NodeJS · GitBook", Ediolot-cookies.herokuapp.com, 2016. [En línea]. Disponible en: <https://ediolot-cookies.herokuapp.com/chapter1/NodeJS.html>. [Accesado: 16- Sep- 2021].
- [49] "Qué es Node.js y por qué debería usarlo", Kinsta, 2021. [En línea]. Disponible en: <https://kinsta.com/es/base-de-conocimiento/que-es-node-js/>. [Accesado: 16- Sep- 2021].
- [50] "MySQL: ¿Qué es? Características, Ventajas y Desventajas", Hostingpedia.net, 2019. [En línea]. Disponible en: <https://hostingpedia.net/mysql.html>. [Accesado: 16- Sep- 2021].
- [51] "MariaDB vs MySQL: ¿cuál debo elegir? - Guiadev". Guiadev. Disponible en: <https://guiadev.com/mariadb-vs-mysql-cual-debo-elegir/> [Accesado: 16- Sep- 2021].
- [52] "Ventajas y desventajas de MySQL, oracle, visual foxpro y access". DISEÑA BASES DE DATOS OFIMATICAS. Disponible en: <https://mape309site.wordpress.com/2017/11/15/ventajas-y-desventajas-de-mysql-oracle-visual-foxpro-y-access/> [Accesado: 16- Sep- 2021].
- [53] "Eclipse IDE: Principales características – Educacion IT". (2014, 16 de enero). Educacion IT – Centro de Capacitación y Desarrollo Profesional. [En línea]. Disponible en: <https://blog.educacionit.com/2014/01/16/eclipse-ide-principales-caracteristicas> [Accesado: 17- Sep- 2021]
- [54] Medina, E. (2018, 19 de febrero). "Thonny es un IDE sencillo que te ayudará a aprender Python» MuyLinux". MuyLinux. [En línea]. Disponible en: <https://www.muylinux.com/2018/02/19/thonny-ide-python/> [Accesado: 17- Sep- 2021]
- [55] "Sensor de temperatura infrarrojo MLX90614". Naylamp Mechatronics - Perú. Disponible en: <https://naylampmechatronics.com/sensores-temperatura-y-humedad/330-sensor-de-temperatura-mlx90614.html> [Accesado: 17- Sep- 2021].
- [56]Melexis. (2019, 13 de septiembre). "Datasheet Melexis MLX90614". [En línea]. Disponible en: <https://mel-prd-cdn.azureedge.net/-/media/files/documents/datasheets/mlx90614-datasheet-melexis.pdf> [Accesado: 17- Sep- 2021]

- [57] "Sensor de temperatura para superficie autoadhesivo RP-RTD". Instrumentos de Medida. Disponible en: <https://www.sensovant.com/productos/temperatura/hvac/articulo/sensor-temperatura-autoadhesivo-superficie-RP-RTD.html> [Accesado: 17- Sep- 2021].
- [58] "Sensor de temperatura digital preciso ADT7422 de $\pm 0.1^{\circ}\text{C}$ ". DigiKey. Disponible en: <https://www.digikey.com.mx/es/product-highlight/a/analog-devices/adt7422-01-c-accurate-digital-temperature-sensor> [Accesado: 17- Sep- 2021].
- [59] MAX30205 Human Body Temperature Sensor" Datasheets MaximIntegrated. [En línea]. Disponible en: <https://datasheets.maximintegrated.com/en/ds/MAX30102.pdf> [Accesado: 17- Sep- 2021].
- [60] "MAX30102 High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health" Datasheets MaximIntegrated. [En línea]. Disponible en: <https://datasheets.maximintegrated.com/en/ds/MAX30205.pdf> [Accesado: 17- Sep- 2021].
- [61] "MAX32664 Ultra-Low Power Biometric Sensor Hub", Datasheets MaximIntegrated. [En línea]. Disponible en: <https://datasheets.maximintegrated.com/en/ds/MAX32664.pdf> [Accesado: 17- Sep- 2021].
- [62] "AD8232 ECG Módulo Monitor de Pulso Cardíaco - UNIT Electronics". (s. f.). UNIT Electronics. [En línea]. Disponible en: <https://uelectronics.com/producto/ad8232-ecg-modulo-monitor-de-pulso-cardiaco/> [Accesado: 17-Sep-2021]
- [63] "Sensor de Pulso Cardiaco - ColdFire Electronica - Todo para Sistemas Embebidos, Arduino, Microcontroladores, Electronica, Sensores, Libros Técnicos, Seminarios, Cursos". ColdFire Electronica - Todo para Sistemas Embebidos, Arduino, Microcontroladores, Electronica, Sensores, Libros Técnicos, Seminarios, Cursos. Disponible en: <https://coldfire-electronica.com/esp/item/231/sensor-de-pulso-cardiaco> [Accesado: 17- Sep- 2021].
- [64] «ESP32 Datasheet». Espressif Systems [Accesado: 17- Sep- 2021].
- [65] Rsuagued. "ARDUINO". Blog de Tecnologías. Disponible en: <https://www3.gobiernodecanarias.org/medusa/ecoblog/rsuagued/arduino/> [Accesado: 17- Sep- 2021].
- [66] J. Pastor, "Raspberry Pi 3 Model B+, análisis: más potencia y mejor WiFi para un miniPC que sigue asombrando", Xataka.com, 2018. [En línea]. Disponible en: <https://www.xataka.com/ordenadores/raspberry-pi-3-model-b-analysis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando>. [Accesado: 14- Oct- 2021].
- [67] J. Beningo. "Desarrollar rápidamente aplicaciones basadas en microcontroladores en tiempo real utilizando MicroPython". Digi-Key. Disponible en: <https://www.digikey.com.mx/es/articles/develop-real-time-mcu-based-applications-micropython> [Accesado: 14- Oct- 2021].
- [68] R. Álvarez Torrico. "Ensamblador Versus C en Microcontroladores". Tecnológico de Bolivia. Disponible en: <https://www.tecbolivia.com/index.php/articulos-y-tutoriales-microcontroladores/6-ensamblador-versus-c-en-microcontroladores> [Accesado: 14- Oct- 2021].
- [69] "IDE Arduino". Aprendiendo Arduino. [En línea]. Disponible en: <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/> [Accesado: 9-Oct-2021].
- [70] J. Guerra Carmenate. "Arduino IDE 2021 en Windows Linux y Mac". Programar fácil con Arduino. [En línea]. Disponible en: <https://programarfácil.com/blog/arduino-blog/arduino-ide/> [Accesado: 9-Oct-2021].
- [71] "Get Started - ESP32 — ESP-IDF Programming Guide latest documentation". (s. f.). Technical Documents | Espressif Systems. [En línea]. Disponible en: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html> [Accesado: 9-Oct-2021]
- [72] No, D. (2019, 18 de abril). "MICROPYTHON ESP32 – Eligiendo un EDITOR DE CODIGO para MicroPython – ESPloradores". ESPloradores – Ayuda e información para DIY MAKERS. [En línea]. Disponible en: https://www.esploradores.com/editorcodigo_micropython-2/ [Accesado: 9-Oct-2021]
- [73] Factores que influyen en la salud: Conócelos todos - Vital Seguro. (s. f.). Vital Seguro. [En línea] Disponible en: <https://www.vitalseguro.com/blog/salud/factores-que-influyen-en-la-salud/> [Accesado: 9-Oct-2021]