



**“Sistema Móvil para Monitoreo de Parámetros de Salud
Transmitidos por un Cinturón Materno Fetal”**

TESIS

QUE PARA OBTENER EL TITULO DE
MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA

ING. ALONSO ÁNGEL SÁNCHEZ MALDONADO

DIRECTORA

M.C. MA. ELENA MARTÍNEZ DURAN

CODIRECTORA

M.C. ROSA DE GUADALUPE CANO ANGUIANO

Villa de Álvarez, Col., a 17 de enero de 2020





EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Colima

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Villa de Álvarez, Colima 21/enero/2020
Oficio No. DEPI 1.2.11/006/2020

**ALUMNO ALONSO ÁNGEL SÁNCHEZ MALDONADO
PASANTE DE LA MAESTRÍA EN SISTEMAS COMPUTACIONALES
PRESENTE**

La División de Estudios de Posgrado e Investigación de acuerdo al procedimiento para la obtención del Título de Maestría de los Institutos Tecnológicos y habiendo cumplido con todas las indicaciones que la comisión revisora hizo a su trabajo profesional denominado "SISTEMA MÓVIL PARA MONITOREO DE PARÁMETROS DE SALUD TRANSMITIDOS POR UN CINTURÓN MATERNO FETAL", por la opción de tesis, que para obtener el grado de Maestro en Sistemas Computacionales será presentado por Usted, tiene a bien concederle la AUTORIZACIÓN de impresión de la tesis citada.

Sin otro particular por el momento, aprovecho la ocasión para enviarle un cordial y afectuoso saludo.

ATENTAMENTE
Excelencia en Educación Tecnológica®

RAMONA EVELIA CHÁVEZ VALDEZ
JEFA DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN



SEP - TecNM
INSTITUTO TECNOLÓGICO
De Colima
DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN

C.p. Archivo.



Av. Tecnológico No. 1, Villa de Álvarez, Col., C.P. 28976
Tel. 01 (312) 314 0933, 314 0683, 312 6393, 312 9920
www.tecnm.mx | www.itcolima.edu.mx





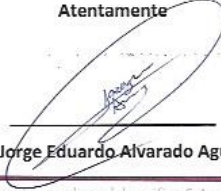
Asunto: Carta de terminación
Guadalajara, Jalisco. 28 de febrero de 2019

Mtra. María Heidi del Pilar Vizcaíno Granados
Jefa del Departamento de Gestión Tecnológica y Vinculación
Instituto Tecnológico de Colima
Presente

Por medio de la presente hago constar, que el Ing. Alonso Ángel Sánchez Maldonado con numero de control G1746003 y alumno de la Maestría de Sistemas Computaciones, termino satisfactoriamente su estancia de tesis profesional con el proyecto "Sistema Móvil para Monitoreo de Parámetros de Salud Transmitidos de un Cinturón Materno Fetal" en el área de sistemas, propiedad de la empresa Soluciones Médicas Del Pacífico S.A. de C.V, durante el periodo de diciembre de 2017 a enero de 2019.

Se extiende la presente a petición del interesado y para los fines legales que a el convengan, quedo de usted para cualquier aclaración.

Atentamente



Dr. Jorge Eduardo Alvarado Aguilar

Soluciones medicas del pacifico S.A. de C.V.
Guadalajara, Jalisco.

DEDICATORIA

Al Instituto Tecnología de Colima, por darme la oportunidad de cursar los estudios de postgrado y compartí el conocimiento de sus maestros.

Al Consejo Nacional de Ciencia y Tecnología, por el apoyo económico que me brindaron para poder estudiar el posgrado.

A mis asesores del proyecto de tesis por brindar su tiempo y dedicación en la realización de este documento, ya que sin su participación no hubiera sido posible culminar este proyecto, así como a los maestros con los que compartí el salón de clases donde recibí el conocimiento de alto nivel propio del programa de posgrado.

A los compañeros estudiantes y maestros por el apoyo, compañerismo y la amistad.

Por último, a mi familia, mis hermanos y mis padres, por ser los pilares de mi vida profesional.

“La ciencia se compone de errores, que, a su vez, son los pasos hacia la verdad.”

Julio Verne.

RESUMEN

Las muertes por sufrimiento fetal en México alcanzaron más de 22,000 muertes tan solo en 2017, siendo el estado de Colima con 128 muertes fetales en el año 2018, la única manera viable de evitar estas muertes es en mayor medida el monitoreo de parámetros como frecuencias cardiacas y de la presión arterial.

El objetivo de esta investigación consiste en desarrollar un sistema móvil para visualizar frecuencias cardiacas y la presión arterial en un dispositivo móvil (Smartphone), para enfrentar problemas en embarazos de alto riesgo.

En el desarrollo de este trabajo se emplearon tecnologías como comunicación Bluetooth, sensores de electrocardiogramas, plataformas open source (Arduino) y lenguajes de programación de la plataforma de desarrollo de Android Studio como Java, Kotlin, C.

Los resultados obtenidos impactan de manera inmediata en casos donde se presentan alteraciones o parámetros de medición fuera de los normales en el embarazo mientras se usa el sistema de monitoreo. se puede concluir que el uso de sistemas móviles o aplicaciones en Smartphones pueden ser de gran ventaja para repartir el procesamiento y censado de sistemas embebidos en el are de la salud.

Palabras clave: Frecuencia cardiaca, presión arterial, Android Studio, Smartphone, sistema móvil, electrocardiogramas, Java.

ABSTRACT

Deaths due to fetal distress in Mexico reached more than 22,000 deaths just in 2017. being the state of Colima with 128 fetal deaths in 2018, the only viable way to avoid these deaths is to a greater extent the monitoring of parameters such as heart rates and blood pressure.

The objective of this research is to develop a mobile system to visualize heart rates and blood pressure in a mobile device (Smartphone), to face problems in high-risk pregnancies.

In the development of this work, technologies such as Bluetooth communication, electrocardiogram sensors, open source platforms (Arduino) and programming languages of the Android Studio development platform such as Java, Kotlin, C. were used.

The results obtained impact immediately in cases where abnormalities or measurement parameters occur outside of normal during pregnancy while using the monitoring system. It can be concluded that the use of mobile systems or applications in Smartphones can be of great advantage to distribute the processing and census of embedded systems in the health area.

Keywords: Heart rate, blood pressure, Android Studio, Smartphone, mobile system, electrocardiograms, Java.

ÍNDICE

ÍNDICE DE TABLAS E ILUSTRACIONES	3
<u>CAPÍTULO 1 INTRODUCCIÓN</u>	<u>7</u>
1.1 NATURALEZA DEL PROBLEMA	9
1.2 EL CONTEXTO DEL PROBLEMA	9
1.3 REVISIÓN DE LA LITERATURA	10
1.4 PLANTEAMIENTO DE LA SOLUCIÓN DEL PROBLEMA A INVESTIGAR	11
1.5 JUSTIFICACIÓN	11
1.6 MOTIVACIÓN.....	12
1.7 OBJETIVOS	12
1.8 METAS	12
1.9 HIPÓTESIS.....	13
1.10 VARIABLE DEPENDIENTE	13
1.11 VARIABLE INDEPENDIENTE	13
1.12 MÉTODOS	13
1.13 ORGANIZACIÓN DE LA TESIS.....	13
<u>CAPÍTULO 2 ESTADO DEL CAMPO DE CONOCIMIENTO.....</u>	<u>15</u>
2.1 INTRODUCCIÓN.....	17
2.2 MARCO HISTÓRICO	17
2.2.1 SITUACIÓN MUNDIAL, NACIONAL Y LOCAL DE LAS MUERTES DURANTE EL EMBARAZO.....	17
2.2.2 FRECUENCIA CARDIACA EN EL EMBARAZO.....	18
2.2.3 PRESIONA ARTERIAL MATERNA.....	19
2.2.4 ELECTROCARDIOGRAMAS MÓVILES	19
2.3 MARCO CONTEXTUAL.....	20
2.3.1 INVESTIGACIONES RELACIONADAS.....	20
2.4 MARCO TEÓRICO.....	21
2.4.1 EL COMPLEJO QRS.....	21
2.4.2 SISTEMA MÓVIL.....	21
2.4.3 SMARTPHONE	22
2.4.4 SISTEMA OPERATIVO ANDROID.....	22
2.4.5 ANDROID SDK.....	23
2.4.6 ANDROID STUDIO	23
2.4.7 COMUNICACIÓN BLUETOOTH.....	23

2.4.8 CONSIDERACIONES PARA DESARROLLAR.....	24
---	----

CAPÍTULO 3 MÉTODOS EMPLEADOS 25

3.1 INTRODUCCIÓN.....	27
3.2 METODOLOGÍA DE DESARROLLO.....	27
3.3 METODOLOGÍA MODELO ESPIRAL.....	28
3.3.1 MODELO ESPIRAL.....	28
3.3.2 MODELO ORIGINAL DE BOEHM.....	29
3.3.3 MODELO TÍPICO DE SEIS REGIONES.....	29
3.3.4 MODELO WINWIN	29
3.4 ACTIVIDADES ESTRATÉGICAS DEL MODELO ESPIRAL	30

CAPÍTULO 4 DESARROLLO DE LA PROPUESTA..... 35

4.1 INTRODUCCIÓN.....	37
4.2 DESARROLLO DE LAS ETAPAS DE LA METODOLOGÍA	37
COMUNICACIÓN	37
PLANEACIÓN.....	37
MODELADO.....	37
CONSTRUCCIÓN	38
DESPLIEGUE	38
4.3 MODELO CONCEPTUAL	38
4.4 DIAGRAMA DE ACTIVIDADES CON RESPONSABILIDADES.....	40
4.5 MODELO DE REQUISITOS.....	40
4.6 REQUISITOS FUNCIONALES.....	41
4.6.1 REQUISITOS SOFTWARE Y HARDWARE	42
4.7 MODELO DE CASOS DE USO.....	44
4.8 DIAGRAMA DE SECUENCIA	46
4.9 DISEÑO DEL ICONO DE APLICACIÓN.....	48
4.10 DISEÑO DE INTERFACES	48
4.10.1 DISEÑO DE INTERFAZ: PANTALLA DE BIENVENIDA.....	48
4.10.2 DISEÑO DE INTERFAZ: CONEXIÓN CON EL CINTURÓN	49
4.10.3 DISEÑO DE INTERFAZ; VISUALIZACIÓN DE DATOS.....	49
4.10.4 DISEÑO DE INTERFAZ: INFORMACIÓN DE PACIENTE	51
4.10.5 DISEÑO DE INTERFAZ: INFORMACIÓN DE MEDICAMENTOS	52
4.10.6 DISEÑO DE INTERFAZ: CONSULTAS PRÓXIMAS.....	52
4.10.7 DISEÑO DE INTERFAZ: REACCIÓN A ALERTAS.....	53
4.11 MODELO DE CLASES	53
4.12 DIAGRAMA DE DATOS	54
4.13 CODIFICACIÓN.....	55
4.14 IMPLANTACIÓN EN EL PROYECTO	60

4.15	DESARROLLO DE PROTOTIPOS.....	66
	PRIMER PROTOTIPO.....	66
	SEGUNDO PROTOTIPO.....	67
	TERCER PROTOTIPO.....	67
	CUARTO PROTOTIPO.....	67
	QUINTO PROTOTIPO.....	67
4.16	PRUEBAS.....	67
<u>CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES.....</u>		69
5.1	INTRODUCCIÓN.....	71
5.2	RESULTADOS.....	71
5.3	CONCLUSIONES Y TRABAJO FUTURO.....	71
<u>BIBLIOGRAFÍA.....</u>		72
<u>ANEXOS.....</u>		ERROR! BOOKMARK NOT DEFINED.

Índice de Tablas e ilustraciones

Ilustración 1	Monitor fetal externo	19
Ilustración 2	Representación del complejo QRS extraído de (Das et al., 2018).....	21
Ilustración 3	Modelo espiral	33
Ilustración 4	Estructura del sistema.....	39
Ilustración 5	Estructura del sistema.....	40
Ilustración 6	Diagrama de actividades.....	40
Ilustración 7	Modelo de casos de uso número 1.....	44
Ilustración 8	Modelo de casos de uso 2	45
Ilustración 9	Modelo de casos de uso 3	46
Ilustración 10	Diagrama de secuencia del sistema	47
Ilustración 11	Icono de aplicación	48
Ilustración 12	Diseño de pantalla bienvenida.....	49
Ilustración 13	Diseño de pantalla conexión.....	49
Ilustración 14	Diseño de pantalla frecuencias cardiacas.....	50
Ilustración 15	Diseño de pantalla presión arterial.....	51
Ilustración 16	Diseño de pantalla calendario de presión arterial.....	51

Ilustración 17 Diseño de pantalla información.....	52
Ilustración 18 Diseño de pantalla medicamentos.	52
Ilustración 19 Diseño de pantalla consulta.....	53
Ilustración 20 Diseño de pantalla de alertas.....	53
Ilustración 21 Modelo de clases.....	54
Ilustración 22 Diagrama de base de datos.	55
Ilustración 23 Ubicación de Clases.	56
Ilustración 24 Ejemplo de clase FCFFragment.....	57
Ilustración 25 Ejemplo de archivo XML.	58
Ilustración 26 Dispositivo emulador del cinturón fetal.....	59
Ilustración 27 Archivo JSON de datos básicos de paciente.	60
Ilustración 28 Pantalla de bienvenida.....	61
Ilustración 29 Pantalla de conexión.	61
Ilustración 30 Pantalla de menú lateral.	62
Ilustración 31 Pantalla de visualización de datos.	62
Ilustración 32 Pantalla de visualización de datos (2).....	63
Ilustración 33 Pantalla de historial de alertas.	63
Ilustración 34 Pantalla de consultas próximas.....	64
Ilustración 35 Pantalla de medicamentos activos.....	64
Ilustración 36 Pantalla de reacción ha alertas.....	65
Ilustración 37 Pantalla de Google mapas.....	65
Ilustración 38 Pantalla de información de paciente.....	66
Ilustración 39 Pantalla de compartir estado.....	66
Ilustración 40 onda R (Ximena Rosana Rivera G., 2009)	67
Tabla 1 Calendarización de las actividades del proyecto	37
Tabla 2 Requerimientos funcionales.	41
Tabla 3 Requerimientos no funcionales.....	43

Capítulo 1

Introducción

1.1 Naturaleza del problema

Las muertes por sufrimiento fetal son un problema de salud pública. En 2017, conforme a las cifras del INEGI, se presentaron 22,336 muertes fetales a nivel nacional (Geografía, 2017). Los números son alarmantes, y si a esto le agregamos que no se cuenta con tecnología eficiente para prevenir que esta situación continúe, nos encontramos ante un círculo vicioso que perpetúa esta realidad, por lo que se considera imperativo actuar para cambiar las estadísticas. A nivel nacional la mayoría de las muertes ocurren después de la semana 28, (Geografía, 2017), aunque de acuerdo con los datos de la Secretaría de Salud de Jalisco (SSJ), la mayoría de las defunciones se presentaron entre la semana 22 y 36. Una de las causas más relevantes de las muertes fetales es la hipoxia fetal, resultado de la falta oxígeno que ocasiona cambios bioquímicos y funcionales particularmente notorios en el sistema nervioso central, o insuficiencia prenatal la cual está ligada a problemas de irrigación sanguínea, desordenes sanguíneos y vasculares de la madre que a su vez puede provocar nacimientos prematuros. (D. Astivia, 2014)

El diagnóstico anticipado de la enfermedad cardíaca durante la etapa de gestación del feto es uno de los retos de la medicina actual. La evaluación del electrocardiograma fetal es, por tanto, crucial para diagnosticar diversas anomalías cardíacas. No obstante, extraer el electrocardiograma fetal para su posterior valoración es una tarea complicada debido a diversos factores que abarcan desde las técnicas de adquisición del mismo y afección de factores fisiológicos, hasta la posible comorbilidad entre diversas patologías. (Sarmiento et al., 2010a).

1.2 El contexto del problema

El monitoreo de la frecuencia cardíaca fetal (FCF) antes del intraparto se ha convertido en una pieza fundamental de la obstetricia moderna, el empleo de la FCF para determinar el estado fetal se ha utilizado desde hace casi 300 años, (Dueñas-García & Díaz-Sotomayor, 2011) esto impulsa el desarrollo de un dispositivo para el monitoreo de la FCF a manera de evitar más muertes en el periodo de gestación. La FCF puede obtenerse mediante la colocación de electrodos sobre el abdomen y el tórax de la madre, sin embargo, el problema radica en que la señal se mezcla con otras señales biomédicas, como la del electrocardiograma materno, la respiración, la actividad estomacal y las contracciones del útero, y por interferencias externas provocadas por la red eléctrica o el ruido térmico. (Sarmiento et al., 2010b)

Hoy en día existen diversos dispositivos médicos (electrocardiogramas fetales) que ayudan a mejorar la calidad de salud humana en la detección y prevención de múltiples enfermedades, sin embargo, dichos aparatos resultan poco prácticos; ya que por su gran tamaño y sus elevados precios únicamente las instituciones y hospitales pueden adquirirlos. (Caballero, Jiménez, Sánchez, & López, 2017) Los dispositivos móviles (Smartphone) son una alternativa para complementar estos aparatos, brindando interfaces más intuitivas, procesadores de datos más potentes, además de la conectividad móvil gracias a sus interfaces de comunicación Wireless, Bluetooth, ZigBee entre otros. Debido al avance en nuevos sistemas operativos móviles, y a la masificación de los sistemas portables como Tablet, Smartphone, PC portátiles y demás; se hace necesario crear equipos biomédicos capaces de conectarse a dichos sistemas y realizar operaciones más complejas que ayuden al médico en el tratamiento, diagnóstico o monitoreo de sus pacientes (De La Cruz, Cuellar, Rojas, Molina, & Robles, 2015).

1.3 Revisión de la literatura

El José Alejandro Díaz Méndez (Dalia & González, 2018) desarrolla un sistema para detectar, con una precisión superior a 90 por ciento, el sufrimiento fetal en mujeres con embarazos de alto riesgo, apoyado en la correlación hemodinámica y el electrocardiograma fetal. En su proyecto utilizan un electrocardiograma fetal y sensores colocados en el vientre de la madre y mediante algoritmos de filtrado y separado de señales se encuentran una precisión de más del 90 por ciento en el electrocardiograma fetal. De esta manera encuentran marcadores evidentes sobre la onda T del complejo QRS.

El artículo de la Fetus Heart Rate Monitoring Using Smart Phone, (Akshada & Patil, 2018) describe un nuevo método para obtener a frecuencia cardíaca fetal [FHR] mediante un teléfono móvil conectado con un pequeño dispositivo Doppler para observar los parámetros fetales del área abdominal, y una banda de muñeca para la presión arterial y los latidos del corazón. Su principal objetivo es dar la opción a las madres de escasos recursos el poder examinar el bienestar del feto y calcular las condiciones actuales de riesgo, especialmente para situaciones críticas de embarazo a través de un pequeño dispositivo móvil integrado.

El artículo titulado Real Time Ambulatory Monitoring System (Radha, 2017), implementa Internet de las cosas (IOT) para el almacenamiento de la información de su modelo de Electrocardiograma, el sistema completo monitorea el estado de la Frecuencia cardíaca en la plataforma Thing Speak de IOT. Principalmente demuestra el monitoreo en tiempo real de la frecuencia cardíaca, frecuencia del pulso y la presión arterial relativa a través especialmente Circuito ECG diseñado y transmisión de datos a un teléfono inteligente a través de Bluetooth.

Por su parte (Caballero et al., 2017) desarrollaron en hardware y programaron un software de un dispositivo a nivel prototipo de bajo costo, ligero y fácil de usar, para que las personas lo usen en casa, y hacer mediciones del comportamiento del corazón, lo desarrollaron sobre un circuito AD8232 en una tarjeta Sparkfun single lead heart rate monitor-AD8232 que asociada a una tarjeta Arduino con comunicación serial a la computadora o por bluetooth hacía un dispositivo inteligente.

Astivia, Gayt y Luna (2014) describen un prototipo de un sistema de registro de ECG abdominal conformado de tres etapas, un Front End analógico para la adquisición y conversión analógica utilizando el circuito ADS1292R, un control y procesamiento con un microcontrolador TM4C129CXNCZAD y el despliegue y configuración en una pantalla táctil. Su objetivo fue lograr una visualización en tiempo real de la señal electrocardiográfica materna-fetal a través de la colocación de electrodos sin gel.

Está claro que existen diferentes modelos para obtener una frecuencia cardiaca y una presión arterial, sin embargo, las nuevas tendencias en dispositivos móviles y a las tecnologías de internet de las cosas (IOT), da la posibilidad de desarrollar aplicaciones y sistemas de monitoreo de la salud en pacientes embarazadas para un dispositivo desarrollado en el país.

1.4 Planteamiento de la solución del problema a investigar

Desarrollar un sistema de monitoreo de señales inalámbricas por medio de una interface Bluetooth para un dispositivo médico ambulatorio no invasivo que monitorea la frecuencia cardiaca fetal, la frecuencia cardiaca materna y la presión arterial con resultados altamente confiables en cualquier ambiente. En casos donde las mediciones expresen valores fuera de parámetros normales se generará un historial para el médico, o en casos extremos se contactará de inmediato con el personal de salud a cargo.

1.5 Justificación

El sistema móvil se conectará con el dispositivo de medición de frecuencia cardiaca, presión arterial y frecuencia cardiaca fetal se desarrollará para el sistema operativo Android por la facilidad para usar recursos del Smartphone y la confiabilidad que ofrece una aplicación nativa.

1.6 Motivación

La principal motivación es desarrollar un sistema móvil para la empresa SOLUCIONES MEDICAS DEL PACIFICO relacionado a su sistema de monitor materno fetal, solucionando así la necesidad de hacer su producto más llamativo para el sector internacional. El desarrollo enfocado en proveer un monitoreo en tiempo real hace posible su introducción haciendo posible reunir tres mediciones vitales para la salud en el embarazo en un solo sistema móvil.

1.7 Objetivos

Objetivo general:

Desarrollar un sistema móvil para visualización de las frecuencias cardiacas y presión arterial en un dispositivo móvil, dicha información se generará de un dispositivo médico con una interface de comunicación Bluetooth, para enfrentar problemas cardiovasculares en el embarazo.

Objetivo específico:

- Desarrollar un sistema que se comunique por una interface Bluetooth.
- Crear una interface de visualización de la información.
- Añadir alertas en caso de presentarse datos fuera de rango.
- Agregar una comunicación con el personal de salud.

1.8 Metas

1. Conectar una aplicación con el prototipo mediante Bluetooth.
 2. Obtener información de la frecuencia cardiaca.
 3. Obtener información de la frecuencia cardiaca fetal.
 4. Obtener información de la presión arterial.
 5. Guardar un promedio diario de los datos.
 6. Desarrollar las alertas para dichas mediciones.
 7. Visualizar en la aplicación la información de la paciente y el médico.
 8. Mostar la información por medio de gráficas para el usuario.
 9. Notificar posibles tareas para la embarazo en caso de una alerta.
-

1.9 Hipótesis

“Un sistema móvil de alertas del monitoreo materno fetal disminuye en un 50% el tiempo de atención de la paciente como reacción más inmediata a un problema cardiovascular durante un embarazo de alto riesgo”.

1.10 Variable dependiente

Los posibles problemas cardiovasculares que enfrenta una mujer durante la etapa del embarazo y la dificultad económica para obtener un dispositivo de monitoreo.

1.11 Variable independiente

La falta de un monitoreo de la presión arterial y la frecuencia cardíaca fetal.

1.12 Métodos

Para el desarrollo del proyecto se implementará la metodología de desarrollo de software en Espiral. Debido a que el modelo se desenvuelve en una serie de entregas evolutivas, donde durante las primeras iteraciones se pueden presentar prototipos y en interacciones posteriores versiones cada vez más completas del sistema (PRESSMAN, 2010).

1.13 Organización de la tesis

El presente trabajo consta de seis capítulos:

Capítulo 1: Introducción, correspondiente al contenido desarrollado hasta el momento donde se expone el problema a solucionar y un estado general de investigación, así como la solución propuesta con objetivos, metas y metodología.

Capítulo 2: Estado del campo de conocimiento, aquí se describen los sistemas existentes en cardiología, y aplicaciones con comunicación Bluetooth de alta confiabilidad.

Capítulo 3: Métodos empleados, se define la metodología que se utilizó en el desarrollo del sistema móvil.

Capítulo 4: Desarrollo, el proceso de creación del proyecto con sus diversas etapas.

Capítulo 5: Resultados, se expone las funcionalidades que logro cubrir el sistema.

Capítulo 6: Conclusiones, se menciona las conclusiones finales del proyecto de investigación y el trabajo que se puede lograr a futuro.

Capítulo 2

Estado del campo de conocimiento

2.1 Introducción

En este capítulo se hablará del estado del campo de conocimiento, identificando el marco histórico, los conceptos básicos con respecto a la frecuencia cardíaca y la presión arterial.

2.2 Marco Histórico

2.2.1 Situación mundial, nacional y local de las muertes durante el embarazo

Como se menciona en el capítulo 1, las muertes por sufrimiento fetal son un grave problema de salud pública, Tan solo en 2017 se presentaron 22,336 muertes fetales a nivel nacional (Geografía, 2017), aunque Colima se encuentra en los estados con menos defunciones con un total de 128 en el año 2018 que comprende al 1.1% de los nacimientos, contraste a estados con más densidad de población, donde oscila desde las 4700 muertes en el Estado de México o el 1.7 % de nacimientos , a 1300 en Jalisco con poco menos del uno por ciento de nacimientos, este último es la sede del proyecto de investigación. Los datos del INEGI a nivel nacional también reflejan que la tasa de defunciones se refleja mayormente en las semanas 22 a 36.

El control de la frecuencia cardíaca fetal es un procedimiento que se utiliza para evaluar el bienestar del feto mediante la determinación de la frecuencia y el ritmo de los latidos del corazón del feto. Durante la última etapa del embarazo y el trabajo de parto, el médico puede optar por controlar la frecuencia cardíaca fetal además de otras funciones (Health, n.d.). La frecuencia cardíaca fetal promedio se encuentra entre 110 y 160 latidos por minuto, y puede variar entre cinco y 25 latidos por minuto. Esta frecuencia puede cambiar como respuesta del feto ante las condiciones intrauterinas. Una frecuencia o ritmo cardíaco fetal anormal puede indicar que el feto no está obteniendo suficiente oxígeno o que hay otros problemas (Sarmiento et al., 2010a).

El pulso mide la frecuencia cardíaca, esto es, es el número de contracciones del corazón o pulsaciones por unidad de tiempo. Pero la frecuencia cardíaca no es igual de una persona a otra. El valor medio de pulsaciones consideradas normales en los adultos es de 60 a 100 pulsaciones por minuto (en reposo) y entre ese margen estarán las pulsaciones de una mujer, embarazada o no (R. Marín Iranzo, M. Gorostidi Pérez, 2012). Las mujeres embarazadas generalmente notan un aumento en la frecuencia del pulso en aproximadamente 10-20 latidos. Entonces, es probable que al final del embarazo se haya pasado de 60-70 pulsaciones por minuto a 80-90. Recuerda que un pulso superior a 130-140 o inferior a 50 es peligroso para el feto, por lo que hay que acudir al médico.

La presión arterial es la presión que ejerce la sangre contra las paredes de las arterias al circular hacia todas las partes del cuerpo (R. Marín Iranzo, M. Gorostidi Pérez, 2012). Las cifras de tensión normales son 120/80 mmHg, y se considera que existe hipertensión cuando las cifras de tensión arterial son iguales o superiores a 140/90 mmHg.

2.2.2 Frecuencia Cardíaca en el embarazo

Según los autores (Astivia, Gayt, & Luna, 2014) La frecuencia cardíaca fetal (FCF) es normal la mayor parte del tiempo, incluso en los embarazos con alto riesgo de prematuridad. Sin embargo, señalan que detectar una variación de la FCF demuestra ser un mecanismo muy útil para evaluar el bienestar del feto, ya que la variación de este parámetro puede ser un indicio de algunos padecimientos. Los autores marcan muy importante durante un embarazo de alto riesgo el monitoreo constante a fin de reducir la incidencia de mortalidad fetal y/o mortandad. Así mismo sugieren el uso de un sistema basado en electrocardiografía abdominal por ser un método pasivo a diferencia del ultrasonido Doppler, que irradia al feto.



Ilustración 1 Monitor fetal externo.

2.2.3 Presiona arterial materna

Otro factor importante en las complicaciones médicas durante el embarazo es la hipertensión arterial (HTA), según el artículo Hipertensión arterial y embarazo (R. Marín Iranzo, M. Gorostidi Pérez, 2012) mencionan la hipertensión presente en un 8-10% de todas las gestaciones y que es la principal causa de mortalidad materna en el mundo. Catalogan la definición de HTA como la misma que la admitida fuera de la gestación: presión arterial sistólica (PAS) mayor de 140 mmHg y/o presión arterial diastólica (PAD) mayor de 90 milímetros de mercurio (mmHg) en dos o más ocasiones consecutivas y separadas por un período de cuatro-seis horas. La HTA durante la gestación se considera grave si la PAS es mayor de 160-170 mmHg y/o la PAD es mayor de 110 mmHg.

2.2.4 Electrocardiogramas móviles

En el artículo de (Maldonado, Angel, Elena, Cano-anguiano, & Guadalupe, 2018) se enumera una cantidad de dispositivos disponibles con funcionalidades para monitorizar la Frecuencia Cardíaca fetal, materna, presión arterial y en algunos dispositivos más parámetros. Entre ellos el más llamativo por ser un pequeño hardware respaldado por un Smartphone es el sistema Shell (Bellabeat, 2016) que consta de un gadget parecido a una concha que se coloca sobre la mujer embarazada, brinda una forma de escuchar los latidos del bebé, combina el método del cuerno o cono de Pinard, utilizando para captar latidos cardíacos de un bebé de forma natural, Shell trabaja filtrando el ruido de fondo, artefactos y amplificando los latidos del corazón del bebé para poder reproducirlos después.

2.3 Marco Contextual

2.3.1 Investigaciones relacionadas

En la tesis de (Grado, Manuel, & Vera, 2015) desarrolla una aplicación para dispositivos con el sistema operativo Android, que sea capaz de registrar un paciente dentro de la red del sistema Promeinfo, y una vez registrado, utilizando los archivos capturados por el dispositivo mostrar una gráfica de la actividad del corazón del paciente, y utilizando esta misma información, se encuentre en la capacidad de enviar esta información para el registro del estado de salud del paciente, realizando envíos de información de forma periódica.

El sistema es muy completo implementando las alertas en caso de emergencias, sin embargo, no tiene una compatibilidad o funcionalidad de un Electrocardiograma fetal.

El Centro De Ingeniería Y Desarrollo Industrial obtuvo una patente (RANGEL BREÑA, ARCINIEGA MONTIEL, & SANTIAGO RAMIREZ, 2017) sobre su proyecto de un método para la adquisición, filtrado y caracterización de la señal cardiaca fetal, en una entrevista (Valencia, 2018) se comenta que el dispositivo integral brinda mayor confiabilidad y precisión a los dispositivos de monitoreo para la vigilancia materno-fetal desarrollados por este centro público de investigación. El objetivo de su patente es un método para separar los latidos maternos y fetales con el objetivo de obtener la señal a través del uso de un grupo de electrodos puestos sobre el abdomen de la madre.

En base a esta investigación se podría asimilar la técnica de filtrado de señales para su correcta interpretación en un sistema móvil.

Recientemente se publicó el artículo "A Fetal ECG Monitoring System Based on the Android Smartphone" (Yuan, Yuan, Zhou, Bai, & Wu, 2019) donde se propone el uso de un Smartphone basado en el Sistema operativo Android, diseñan un colector de ECG fetal portátil de baja potencia que envía datos en tiempo real por medio de Bluetooth a la aplicación de Android. La aplicación integró el algoritmo rápido de punto fijo para el análisis de componentes independientes (FastICA) y el algoritmo de entropía de muestra, en aras de la extracción en tiempo real de señales de ECG fetales de la madre. La frecuencia cardíaca fetal se calculó utilizando las señales de ECG fetales extraídas. Los resultados experimentales mostraron que el algoritmo FastICA puede extraer un ECG fetal claro, y la entropía de la muestra puede determinar correctamente el canal donde se encuentra el ECG fetal. El sistema de monitoreo de ECG fetal propuesto puede ser factible para el monitoreo no invasivo en tiempo real de los ECG fetales.

Aunque este trabajo carga la mayoría del trabajo de cálculo de señal en el dispositivo Android, no cuenta con mediciones de frecuencia cardíaca materna y de presión arterial. Sin embargo, es un buen proyecto de monitoreo la implantación de alertas en caso de rangos fuera de parámetros normales lo haría más llamativo para su comercialización.

2.4 Marco Teórico

2.4.1 El complejo QRS

En el análisis de un EKG se miden dos intervalos PR y QRS sin embargo nos enfocaremos en el segundo. El intervalo QRS se mide desde el principio de la primera onda del QRS, en el punto en que se eleva a partir de la línea isoelectrónica, hasta el final de la última onda del QRS, donde se encuentra con la línea isoelectrónica. Cuente a lo largo del eje horizontal intervalos de 0.04s hasta que obtenga la distancia entre los dos puntos; este es el intervalo QRS en segundos.(Educaci et al., 2015) Los valores normales para el intervalo QRS son de 0.04s a 0.11s.

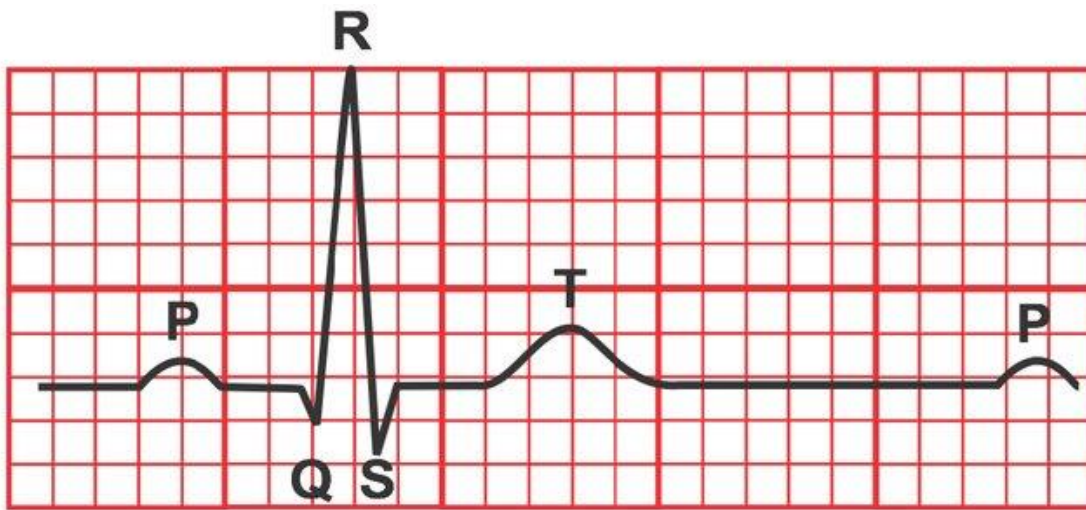


Ilustración 2 Representación del complejo QRS extraído de (Das et al., 2018).

2.4.2 Sistema móvil

Un sistema móvil o también llamado aplicación (también llamada app) es simplemente un programa informático creado para llevar a cabo o facilitar una tarea en un dispositivo informático. Cabe destacar que, aunque todas las aplicaciones son programas, no todos los programas son aplicaciones. Existe multitud de software en el mercado, pero sólo se denomina así a aquel que ha sido creado con un fin determinado, para realizar tareas concretas. No se consideraría una aplicación, por ejemplo, un sistema operativo, ni una suite, pues su propósito es general. En general, una aplicación se diseña para ser usada con uno o varios sistemas operativos, siendo hasta cierto punto dependiente de estos. Pero también existen aplicaciones diseñadas en exclusiva para el uso con un programa o tipo de programas.(por el Bachiller, Robertho Luty Artica Navarro Asesor, & Luis Honorato Pita Astengo, 2014)

2.4.3 Smartphone

De acuerdo con (Baz, Ferreira, Álvarez, & García, 2011) un “Smartphone” es un dispositivo eléctrico que funciona como un móvil con características muy similares a las de un ordenador personal. Una característica de casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad.

Un “Smartphone” (teléfono inteligente en español) es un dispositivo electrónico que funciona como un teléfono móvil con características similares a las de un ordenador personal. Es un elemento a medio camino entre un teléfono móvil clásico y una PDA ya que permite hacer llamadas y enviar mensajes de texto como un móvil convencional pero además incluye características cercanas a las de un ordenador personal. Una característica importante de casi todos los teléfonos inteligentes es que permiten la instalación de programas para incrementar el procesamiento de datos y la conectividad. Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador o por un tercero. Los teléfonos inteligentes se distinguen por muchas características, entre las que destacan las pantallas táctiles, un sistema operativo, así como la conectividad a Internet y el acceso al correo electrónico.(Baz et al., 2011)

2.4.4 Sistema operativo Android

Android es un sistema operativo móvil basado en Linux y Java que ha sido liberado bajo la licencia Apache versión 2. El sistema busca, nuevamente, un modelo estandarizado de programación que simplifique las labores de creación de aplicaciones móviles y normalice las herramientas en el campo de la telefonía móvil. Al igual que ocurriera con

Symbian, lo que se busca es que los programadores sólo tengan que desarrollar sus creaciones una única vez y así ésta sea compatible con diferentes terminales.

Una de las claves de la popularidad de Android es que, como Linux, es una plataforma de código abierto, lo que permite a fabricantes, operadores y desarrolladores dar mayor funcionalidad a sus Smartphone (Báez, 2012).

2.4.5 Android SDK

El SDK de Android incluye numerosas y completas API's para facilitar el desarrollo. Algunas de las características más relevantes son; Licencias, distribución y desarrollo gratuitos, tampoco hay procesos de aprobación del software. Acceso al hardware de Wifi, GPS, Bluetooth y telefonía, permitiendo realizar y recibir llamadas y SMS. Control completo de multimedia, incluyendo la cámara y el micrófono. APIs para los sensores: acelerómetros y brújula. Almacenes de datos compartidos, SQLite, acceso a SD Card. Aplicaciones y procesos en segundo plano. Widgets para la pantalla de inicio (escritorio). Integración de los resultados de búsqueda de la aplicación con los del sistema. Uso de mapas y sus controles desde las aplicaciones. Aceleración gráfica por hardware, incluyendo OpenGL ES 2.0 para los 3D. Muchas de estas características ya están, de una manera o de otra, para los SDK de otras.

2.4.6 Android Studio

Android Studio es el IDE oficial de Android. Está diseñado para que Android pueda acelerar el desarrollo y te permita crear las apps de mejor calidad para todos los dispositivos de Android. Ofrece herramientas personalizadas para programadores de Android. Se incluyen herramientas completas de edición, depuración, pruebas y perfilamiento de códigos. Android Studio, basado en IntelliJ IDEA, proporciona el menor tiempo de respuesta en tu flujo de trabajo de codificación y ejecución. Android Studio ofrece automatización de compilaciones, administración de dependencias y configuraciones de compilación personalizables. Puedes configurar tu proyecto de modo que se incorporen bibliotecas locales y alojadas, y definir variantes que incluyan código y recursos diferentes, además de aplicar configuraciones de reducción de código y firma de apps. ("Funciones de Android Studio," n.d.)

2.4.7 Comunicación Bluetooth

La plataforma de Android incluye compatibilidad con la pila de red Bluetooth, la cual permite que un dispositivo intercambie datos de manera inalámbrica con otros

dispositivos Bluetooth. El Frameworks de la aplicación proporciona acceso a la funcionalidad Bluetooth mediante las Android Bluetooth API. Estas API permiten a las aplicaciones conectarse de manera inalámbrica con otros dispositivos Bluetooth y habilitan las funciones inalámbricas punto a punto y de multipunto.

Con las Bluetooth API, una aplicación de Android puede realizar lo siguiente: buscar otros dispositivos Bluetooth, consultar el adaptador local de Bluetooth en busca de dispositivos Bluetooth sincronizados, transferir datos hacia otros dispositivos y desde estos, administrar varias conexiones. (Bluetooth,2018)

2.4.8 Consideraciones para desarrollar

El desarrollo para dispositivos móviles y embebidos requiere que el programador tenga especial cuidado en determinados aspectos. El sistema operativo no puede encargarse de controlar todo porque limitaría demasiado al programador, así que determinados aspectos como ahorrar CPU y memoria son responsabilidad del programador en la mayoría de los casos. Las limitaciones de hardware que el dispositivo impone suelen ser: Pequeña capacidad de procesamiento, memoria RAM limitada, memoria permanente de poca capacidad, pantallas pequeñas de poca resolución transferencias de datos costosa (en términos de energía y económicos) y lenta, inestabilidad de las conexiones de datos, batería muy limitada y la necesidad de terminar la aplicación en cualquier momento. A partir de estas limitaciones, las consideraciones que el desarrollador debe tener son: ser eficiente en cuanto a bucles y cálculos; asumir poca memoria y optimizar el almacenamiento al máximo, eliminando siempre los datos que ya no van a ser necesarios; diseñar para pantallas de distintos tamaños pero sobre todo tener en cuenta las más pequeñas; ahorrar comunicaciones y asumir que serán lentas y que fallarán, lo cual no debe limitar la capacidad de respuesta de la aplicación.(Robledo Fernández & Robledo, 2014).

Capítulo 3

Métodos empleados

3.1 Introducción

En este capítulo se describe los pormenores de la metodología utilizada en el desarrollo de la tesis, detallando los métodos, técnicas, procedimientos y herramientas que se involucraron en las etapas del trabajo.

3.2 Metodología de desarrollo

Una metodología de desarrollo de software es un proceso que busca desarrollar productos o soluciones para problemas planteados por un cliente, en los cuales se debe considerar los costos, planificación, calidad y entregables. Para este proceso es indispensable contar con una comunicación amplia y precisa con el usuario, en la cual se analicen, definan, construyan y manipulen los modelos para un intercambio de información y datos entre los participantes. (OBS Business School, 2019).

En el libro Ingeniería en software de Ian Sommerville (Sommerville, 2005) los modelos para el desarrollo de software pueden agruparse de la siguiente forma:

a) Modelo de cascada:

Las actividades están relacionadas unas a otras de modo que el proceso en su conjunto avanza cuando cada etapa queda definida y el desarrollo continúa con la siguiente etapa. Las acciones principales del desarrollo de un programa de software son representadas como fases de procesos separados tales como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, entre otros.

b) Modelo de desarrollo evolutivo:

En este caso, por el contrario, lo más importante no es la suma de aportes de cada etapa del proceso, sino el hecho de que las actividades de especificación, desarrollo y validación están entrelazadas. El punto de partida es un sistema inicial que se desarrolla de forma rápida y que va evolucionando según la dinámica del propio proyecto y las peticiones de los clientes o destinatarios. Todo el proceso es una continua evolución que sólo se detiene hasta que los objetivos iniciales han sido alcanzados.

c) Modelo de componentes:

Se trata de un modelo especialmente útil en procesos que parten del trabajo que otros han llevado a cabo. El modelo se basa en la adecuación y adaptación de partes del sistema que ya existen, que al final del mismo acaban cobrando un nuevo valor y asumiendo otras funciones.

Estos modelos no son excluyentes, en algunos casos se complementan con el objetivo de cubrir las necesidades del proyecto que con un solo modelo no podrían ser cubiertas.

3.3 Metodología modelo Espiral

Para el desarrollo del proyecto de tesis se implementa la metodología en Espiral de Barry Boehm “*WINWIN*” (PRESSMAN, 2010), considerando que es un tipo de investigación aplicada para resolver una problemática de un caso práctico.

El modelo de ciclo de vida en espiral de B.W. Boehm surge en la década de los 80 para sustituir la solución en fases del “modelo en cascada” con ciclos de experimentación y aprendizaje (ALONSO, MARTINEZ, & SEGOVIA, 2005). El modelo incorpora un nuevo elemento en el desarrollo al cual denomina “análisis de riesgos”. El modelo es la unión del modelo de prototipos y la secuencialidad del clásico, esta mezcla permite construir un ciclo que inicia desde el centro, si el usuario prefiere hacer alguna modificación o mejoría al producto, se evalúan las diferentes alternativas y se crea un nuevo análisis de riesgos, con lo cual se forma una nueva vuelta de espiral, de esta manera se incrementa el número de vueltas hasta que el producto queda terminado. Este modelo ha evolucionado en diferentes propuestas las cuales son:

3.3.1 Modelo espiral

Fue originalmente propuesto por BOEHM en 1976, y es un modelo de proceso de software evolutivo donde se conjuga la naturaleza de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal y secuencia, proporciona potencial para el desarrollo rápido de versiones incrementales del software que no se basan en fases claramente definidas y separadas para desarrollar un sistema, el modelo desarrolla una serie de versiones incrementales, el resultado de las iteraciones producen versiones cada vez más complejas del sistema. El modelo se divide en un número de actividades de marco de trabajo también llamadas Regiones de tareas, cada región está compuesto por un conjunto de trabajo llamado conjunto de tareas que se adaptan a características del proyecto que va a emprenderse.

3.3.2 Modelo original de Boehm

Para este modelo no hay un número definido de iteraciones, pues estas se gestionan por el equipo del proyecto, se compone de 4 sectores; planeación, análisis de riesgo, ingeniería y evaluación. El movimiento de la espiral se amplía con cada iteración en su amplitud radial, esto indica que cada vez se va construyendo versiones sucesivas y más completas del software.

3.3.3 Modelo típico de seis regiones

Este modelo a diferencia del proceso clásico puede adaptarse y aplicarse a lo largo de la vida del software, puede ser considerada una versión alternativa del modelo espiral. Las regiones de tarea que comprenden este modelo son, comunicación con el cliente, planificación, análisis de riesgo, ingeniería, construcción y adaptación, y finalmente evaluación del cliente.

3.3.4 Modelo winwin

El modelo espiral winwin define un conjunto de actividades de negociación al principio de cada paso alrededor de la espiral, dentro de esta actividad de comunicación con el cliente se definen las siguientes actividades: Identificación del sistema, determinación de las condiciones del cliente, negociación de las metas de las etapas con el cliente. Este modelo introduce tres hitos en el proceso, llamados puntos de fijación que ayudan a establecer la completitud de un ciclo alrededor de la espiral y proporcionan la decisión de continuar con el proyecto.

Este modelo espiral contiene las siguientes ventajas, las cuales fueron clave para tomarla como base para la realización de este proyecto;

- El modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora.
- Como el software evoluciona a medida que progresa el proceso, el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos.
- El modelo en espiral permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.

-
- El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto y si se aplica adecuadamente debe reducir los riesgos antes de que se conviertan en problemas.
 - En la utilización de grandes sistemas ha doblado la productividad.

3.4 Actividades estratégicas del modelo espiral

De acuerdo con Pressman (PRESSMAN, 2010) , las cinco actividades del modelo son

Comunicación. Esta actividad es indispensable antes de que comience cualquier trabajo técnico, tiene importancia crítica reunirse con todos los participantes del proyecto, clientes, analistas y desarrolladores, para definir los objetivos generales del software, identificar los requerimientos y detectar las áreas en las que es indispensable una mayor definición. Se busca entender los objetivos de los participantes respecto del proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software.

Planeación. Incluye un conjunto de prácticas administrativas y técnicas que permiten que el equipo de software defina un mapa mientras avanza hacia su meta estratégica y sus objetivos tácticos. Un proyecto de software es un proceso difícil, y la actividad de planeación crea un “mapa” que guía al equipo durante el desarrollo del mismo. El mapa —llamado plan del proyecto de software— define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los recursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades.

Modelado. La creación de modelos permite entender mejor la entidad real que se va a construir. En el desarrollo de un proyecto de software, el modelo debe ser capaz de representar la información que el software transforma, la arquitectura y las funciones que permiten que esto ocurra, las características que desean los usuarios y el comportamiento del sistema mientras la transformación tiene lugar. Los modelos deben cumplir estos objetivos en diferentes niveles de abstracción, en primer lugar, con la ilustración del software desde el punto de vista del cliente y después con su representación en un nivel más técnico.

Para ello se crean dos clases de modelos:

De requerimientos, también conocidos como modelos de análisis, representan los requerimientos del cliente mediante la ilustración del software en tres dominios diferentes: el de la información, el funcional y el de comportamiento. Para ello se aplican los siguientes principios operacionales:

Principio 1. Debe representarse y entenderse el dominio de información de un problema, incluye los datos que fluyen hacia el sistema (usuarios finales, otros sistemas o dispositivos externos), los datos que fluyen fuera del sistema (por la interfaz de usuario, interfaces de red, reportes, gráficas y otros medios) y los almacenamientos de datos que recaban y organizan objetos persistentes de datos (por ejemplo, aquellos que se conservan en forma permanente).

Principio 2. Deben definirse las funciones que realizará el software. Las funciones se describen en muchos y distintos niveles de abstracción, que van de un enunciado de propósito general a la descripción detallada de los elementos del procesamiento que deben invocarse.

Principio 3. Debe representarse el comportamiento del software (como consecuencia de eventos externos).

Principio 4. Los modelos que representen información, función y comportamiento deben dividirse de manera que revelen los detalles en forma estratificada (o jerárquica).

Principio 5. El trabajo de análisis debe avanzar de la información esencial hacia la implementación en detalle. El modelado de requerimientos comienza con la descripción del problema desde la perspectiva del usuario final. Se describe la “esencia” del problema sin considerar la forma en la que se implementará la solución.

De diseño. Representan características del software que ayudan a los profesionales a elaborarlo con eficacia: arquitectura, interfaz de usuario y detalle en el nivel de componente. Para ello se aplican los siguientes principios operacionales:

Principio 1. El diseño debe poderse rastrear hasta el modelo de requerimientos.

Principio 2. Siempre tomar en cuenta la arquitectura del sistema que se va a construir.

Principio 3. El diseño de los datos es tan importante como el de las funciones de procesamiento.

Principio 4. Las interfaces (tanto internas como externas) deben diseñarse con cuidado.

Principio 5. El diseño de la interfaz de usuario debe ajustarse a las necesidades del usuario final. Sin embargo, en todo caso debe resaltar la facilidad de uso.

Principio 6. El diseño en el nivel de componentes debe tener independencia funcional.

Principio 7. Los componentes deben estar acoplados con holgura entre sí y con el ambiente externo.

Principio 8. Las representaciones del diseño (modelos) deben entenderse con facilidad.

Principio 9. El diseño debe desarrollarse en forma iterativa. El diseñador debe buscar más sencillez en cada iteración.

Construcción. Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste. La actividad de construcción incluye un conjunto de tareas de codificación y pruebas que lleva a un software operativo listo para entregarse al cliente o usuario final. La codificación puede ser:

La creación directa de lenguaje de programación en código fuente. La generación automática de código fuente que usa una representación intermedia parecida al diseño del componente que se va a construir, o la generación automática de código ejecutable que utiliza un “lenguaje de programación de cuarta generación”. Las pruebas dirigen su atención inicial al componente, y con frecuencia se denomina prueba unitaria. Otros niveles de pruebas incluyen de integración (realizadas mientras el sistema está en construcción), de validación, que evalúan si los requerimientos se han satisfecho para todo el sistema (o incremento de software), y de aceptación, que efectúa el cliente en un esfuerzo por utilizar todas las características y funciones requeridas.

Despliegue. El software (como entidad completa o como un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación. Incluye tres acciones: entrega, apoyo y retroalimentación.

El despliegue ocurre no una vez sino varias, a medida que el software avanza hacia su conclusión. Cada ciclo de entrega pone a disposición de los clientes y usuarios finales un incremento de software operativo que brinda funciones y características utilizables. Cada ciclo de apoyo provee documentación y ayuda humana para todas las funciones y características introducidas durante los ciclos de despliegue realizados hasta ese momento. Cada ciclo de retroalimentación da al equipo de software una guía importante que da como resultado modificaciones de las funciones, de las características y del enfoque adoptado para el siguiente incremento.

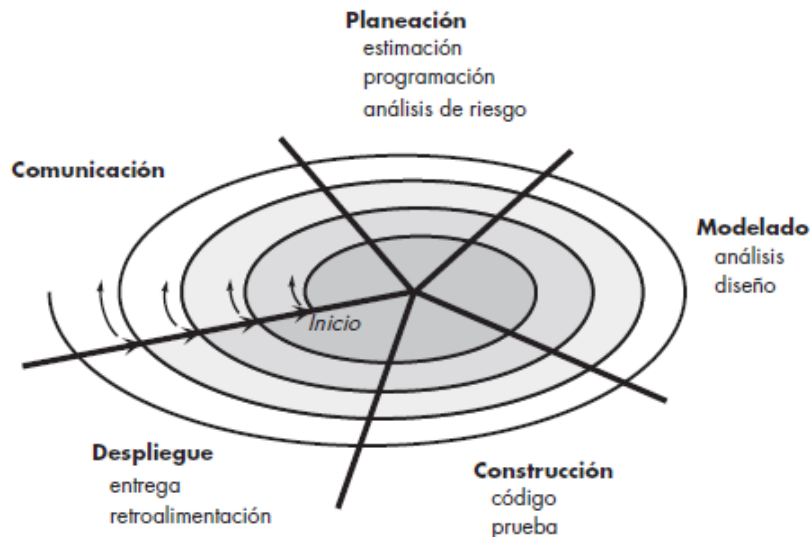


Ilustración 3 Modelo espiral.

El modelo espiral es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada. Tiene el potencial para hacer un desarrollo rápido de versiones cada vez más completas. Boehm (PRESSMAN, 2010) describe el modelo del modo siguiente:

“El modelo de desarrollo espiral es un generador de modelo de proceso impulsado por el riesgo, que se usa para guiar la ingeniería concurrente con participantes múltiples de sistemas intensivos en software. Tiene dos características distintivas principales. La primera es el enfoque cíclico para el crecimiento incremental del grado de definición de un sistema y su implementación, mientras que disminuye su grado de riesgo. La otra es un conjunto de puntos de referencia de anclaje puntual para asegurar el compromiso del participante con soluciones factibles y mutuamente satisfactorias.”

Con el modelo espiral el software se desarrolla en una serie de entregas evolutivas. Con las primare iteraciones lo que se entrega puede ser un modelo o prototipo, y con las iteraciones posteriores se desarrollan versiones cada vez más completas del sistema.

Un modelo en espiral es dividido por el equipo de software en un conjunto de actividades estructurales. Cada una de ellas representa un segmento de la trayectoria espiral ilustrada en la ilustración 3. Con el proceso evolutivo, el equipo de software realiza actividades alrededor de la espiral, partiendo del centro. El riesgo se considera conforme se desarrolla cada revolución. En cada paso se marcan puntos de referencia.

El paso da resultado el desarrollo de una descripción del producto; las vueltas sucesivas se usan para desarrollar un prototipo y, luego, versiones cada vez más sofisticadas del software. Cada paso por la región de planeación da como resultado ajustes en el plan del proyecto. El gerente del proyecto ajusta el número planeado de iteraciones que se requieren para terminar el software (PRESSMAN, 2010).

El modelo espiral puede adaptarse para aplicarse a lo largo de toda la vida del software de cómputo. En el primer circuito alrededor de la espiral quizá represente un “proyecto de desarrollo del concepto” que comienza en el centro de la espiral y continúa por iteraciones múltiples hasta que queda terminado el desarrollo del concepto. El nuevo producto evolucionará a través de cierto número de iteraciones alrededor de la espiral. Más adelante puede usarse un circuito alrededor de la espiral para que represente un “proyecto de mejora del producto”. En esencia, la espiral, cuando se caracteriza de este modo, sigue operativa hasta que el software se retira.

El modelo espiral es un enfoque realista para el desarrollo de sistemas y de software a gran escala (PRESSMAN, 2010). El modelo espiral usa los prototipos como mecanismo de reducción de riesgos, pero, más importante, permite aplicar el enfoque de hacer prototipos en cualquier etapa de la evolución del producto. Mantiene el enfoque de escalón sistemático sugerido por el ciclo de vida clásico, pero lo incorpora en una estructura iterativa que refleja al mundo real en una forma más realista.

Es difícil convencer a los clientes (en particular en situaciones bajo contrato) de que el enfoque evolutivo es controlable (PRESSMAN, 2010). Demanda mucha experiencia en la evaluación del riesgo y se basa en ella para llegar al éxito. No hay duda de que habrá problemas si algún riesgo importante no se descubre y administra.

Capítulo 4

Desarrollo de la propuesta

4.1 Introducción

En este capítulo se presentan las actividades desarrolladas para cada una de las fases de la metodología establecida. Para ello se realizó una calendarización que se presenta en la tabla 1

No. Actividad	Nombre	Inicio	Termino
1	Comunicación	Enero 2018	Enero 2019
2	Planeación	Febrero 2018	Marzo 2019
3	Modelado	Marzo 2018	Marzo 2019
4	Construcción	Abril 2019	Junio 2019
5	Despliegue	Marzo 2019	Julio 2019

Tabla 1 Calendarización de las actividades del proyecto.

4.2 Desarrollo de las etapas de la metodología

Comunicación

La primera fase del proyecto se basó en concretar las necesidades y requerimientos del sistema, por medio de entrevistas entre las partes interesadas, se sentaron las bases de la solución del proyecto que sería el desarrollo de una aplicación móvil que registrase un historial de frecuencias cardiacas fetales complementando la funcionalidad del proyecto de cinturón materno fetal.

Planeación

La planeación aborda elementos para definir el flujo de trabajo, que a su vez permite rastrear el desarrollo de avances conforme la realización del proyecto, las tareas principales o funcionalidades base se desarrollaron en los primeros prototipos. En este modelo es incluido una fase llamada análisis de riesgo la cual va evaluando los riesgos técnicos y otras informaciones relacionadas del proyecto.

Modelado

En algunos casos también llamado Ingeniería, se desarrollan las tareas requeridas para construir una o más representaciones de la aplicación. Elementalmente es el desarrollo de los diseños de clase, bases de datos e interfaces para la construcción de la iteración del sistema.

Construcción

Se mejora y revisa el diseño del componente. El código se genera, revisa, compila y prueba. Las pruebas se mantienen para todas las tareas y resultados de trabajo de importancia por iteración del modelo.

Despliegue

Las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de modelado e implementada durante la etapa de construcción, caracterizada por retroalimentaciones en las entregas de iteraciones.

4.3 Modelo conceptual

Como lo muestra la Ilustración 4 los elementos que componen el sistema del cinturón materno fetal son; el cinturón materno fetal; se compone de una serie de electrodos conectados al vientre y brazos de la madre, un esfigmomanómetro digital, un controlador y puertos USB para cargar la batería interna. Cuenta con conectividad Bluetooth la cual usa para enviar parámetros y alertas al sistema móvil. El sistema móvil está montado en un dispositivo Android recibe los parámetros del cinturón materno fetal para desplegarlos en las diferentes pantallas, reacciona a las alertas para informar al usuario, también se comunica al con el sistema web de la empresa para mostrar datos de la paciente, doctor, medicamentos entre otros. El sistema web de la empresa crea, edita, y vincula los datos del cinturón materno fetal con las de la paciente, también almacena anomalías y alerta al personal médico.

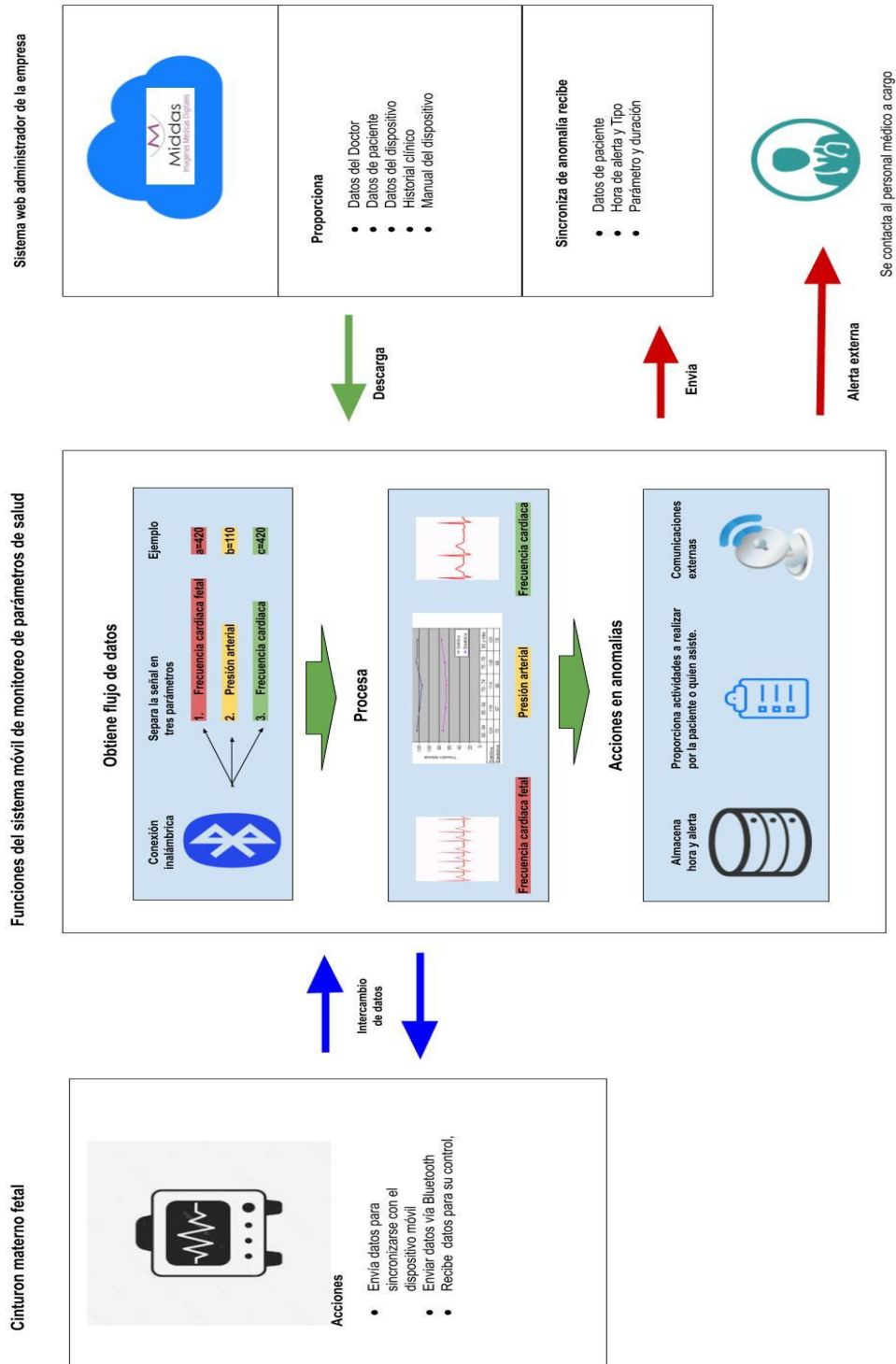


Ilustración 4 Estructura del sistema.

Ilustración 5 Estructura del sistema.

4.4 Diagrama de actividades con responsabilidades

El siguiente diagrama de actividades Ilustración 5, instruye el proceso completo del funcionamiento de monitoreo de las frecuencias cardiacas y alertas dentro del sistema móvil.

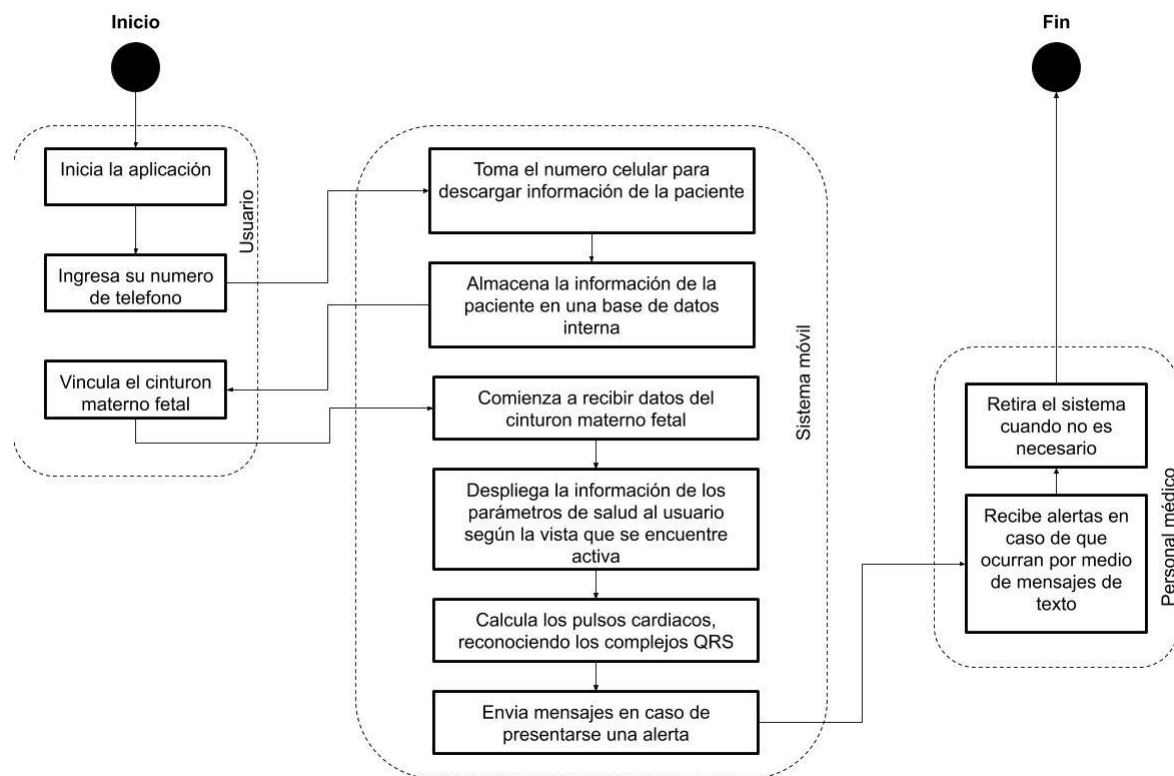


Ilustración 6 Diagrama de actividades.

4.5 Modelo de requisitos

En esta sección se mostrarán los requisitos necesarios para la funcionalidad del sistema móvil, y para su protocolo de comunicación con el cinturón fetal y el sistema web. Dada la naturaleza cambiante de los requerimientos por la metodología espiral se optó por dejar los requerimientos del ultimo entregable.

4.6 Requisitos funcionales

Los requerimientos funcionales se reflejan en la tabla 2.

Tabla 2 Requerimientos funcionales.

Actividades del sistema móvil	
01	Debe existir una pantalla de bienvenida con un campo de texto con el cual el usuario ingresara su número de teléfono, un botón para iniciar la función de COMENZAR y un botón para mostrar el aviso de privacidad.
02	Al presionar el botón COMENZAR, se descargará la información de la paciente, el servidor recibe el número de teléfono, la hora y la llave privada del sistema para autorizar la comunicación y regresa un objeto JSON con la información, una vez descargada debe ser almacenada en el Smartphone.
03	La información del objeto JSON será; -nombre de paciente, dirección, número de teléfono, numero de derechohabiente, fecha de nacimiento, fecha tentativa de concepción, nombre tentativo del bebe, nombre del doctor, número de teléfono del doctor, hospital más cercano, número de teléfono del hospital, numero de cama o consultorio, una lista de medicamentos y notas.
04	Existirá una pantalla para conectarse al cinturón materno fetal, el usuario selecciona el cinturón y comienza la comunicación o vinculación.
05	La pantalla principal del sistema muestra la información de la frecuencia cardiaca fetal.
06	Se necesita un menú lateral para la navegación entre funcionalidades.
Secciones del menú lateral principal	
07	La sección principal debe desplegar la información de frecuencia cardiaca fetal, frecuencia cardiaca materna y presión arterial por separado, y debe estar monitoreando los parámetros normales.
08	El sistema recibirá cada cierto tiempo los datos de las frecuencias cardiacas y la presión arterial.
09	Existirán dos tipos de alertas, bajas y altas; las bajas, se almacenarán en el Smartphone, se notificará al médico por medio de un mensaje de texto desde el sistema móvil, y se mandará el dato de la fecha y el tipo al sistema web.
10	Para las alertas altas, se almacenarán en el Smartphone, se notificará al médico por medio de un mensaje de texto desde el sistema móvil, y se mandara el dato de la fecha y el tipo al sistema web, además se lanzará el mapa de navegación de Google para que la paciente se dirija a su hospital.
11	Existirá una alerta baja de frecuencia cardiaca fetal, cuando los latidos por minuto se encuentren entre 110 a 119 por minuto o menores a 110 y cuando estén sobre 170 o entre 160 a 169, durante 5 minutos.

12	Una alerta alta de la frecuencia cardiaca fetal es cuando la alerta baja sobre pasa los 15 minutos.
13	Existirá una alerta baja de frecuencia cardiaca materna, cuando los latidos por minuto sean menores a 60 o mayores a 100, durante 5 minutos.
14	Otra alerta alta de frecuencia cardiaca materna, cuando los latidos por minuto sean menores a 60 o mayores a 100, durante 15 minutos.
15	La última alerta baja es sobre presión arterial, se presentará si la presión diastólica es menor a 80 o la presión sistólica mayor a 120 milímetros de mercurio, en caso de durar más de 5 minutos.
16	Otra alerta alta sobre presión arterial, si la presión diastólica menor a 80 o la presión sistólica mayor a 120 milímetros de mercurio, en caso de durar más de 15 minutos.
Secciones del menú lateral Historia clínica	
17	Para esta sección se necesita una pantalla donde se despliegan las consultas próximas de la paciente, esta información se debe de actualizar al menos una vez al día, se descarga del sistema web por medio de objetos JSON, cuando exista una consulta en las siguientes 72 horas se notificará al usuario.
18	La información del objeto JSON de consultas próximas será; -Fecha, hora y turno de la consulta, así como el tipo, la especialidad y el médico, también una nota en caso de requerir instrucciones extras para la consulta.
19	La siguiente pantalla desplegara una lista de medicamentos, que se recetaron en las últimas consultas o que estén activas en ese momento, esta información se descargara del sistema web al por medio de objetos JSON.
20	La información del objeto JSON de los medicamentos será; -El nombre del medicamento, la ruta para descarga una imagen, la aplicación, y una descripción.
Secciones del menú lateral Comunicación	
21	Existirá una funcionalidad llamada clínica más cercana a cuál lanzará la aplicación de mapas de Google, para dirigirse a la clínica asignada. La información de la clínica se descargó al principio de la aplicación.
22	Se agrega una funcionalidad, para compartí los latidos actuales del bebe en las aplicaciones social disponibles del Smartphone.
23	Se puede enviar una alerta directa al médico, con un botón, la alerta se enviará por un mensaje de texto, pero no se registrará en el sistema web.
24	Una última pantalla llamada información, muestra la información de la paciente, que se descargó en el JSON el principio.

4.6.1 Requisitos Software y Hardware

También llamados requerimientos no funcionales se muestran en la tabla 3. Estas herramientas son necesarias para la integración del sistema.

Tabla 3 Requerimientos no funcionales.

1	<p>El sistema móvil</p> <ul style="list-style-type: none"> • El sistema debe ser capaz de procesar los datos a 9600 baudios, recibidos por el cinturón por minuto, y mostrarlos en la pantalla. • La conexión se debe de realizar en menos de 5 movimientos en la pantalla. • Los datos que se almacenan en la memoria del Smartphone deben actualizarse al menos cada 20 minutos. • Los permisos necesarios para su funcionamiento no pueden ser desactivados después de ser concedidos. • El sistema debe automatizar una reconexión en caso de pérdida de datos. • El sistema termina su funcionamiento cuando el cinturón deje de enviar datos. • Las interfaces deben ser responsivas a los diferentes tamaños de pantallas de Smartphones. • Las interfaces graficas deben ser claras y bien formadas.
2	<p>Un Smartphone con sistema operativo Android de Google.</p> <ul style="list-style-type: none"> • Debe tener la versión 5.0 (Lollipop) o superior. • Conectividad Bluetooth 3.0 o superior. • Una conexión a internet. • Almacenamiento mínimo de 20 megabytes para la aplicación.
3	<p>El cinturón materno fetal</p> <ul style="list-style-type: none"> • Conectividad Bluetooth 3.0 o superior.
4	<p>El sistema web</p> <ul style="list-style-type: none"> • Con protocolo HTTP servidor para la descarga de los objetos JSON. • Se tiene que efectuar un respaldo en el sistema web mínimo una vez al día.
5	<p>Android Studio</p> <ul style="list-style-type: none"> • Módulo de cálculo de carga de trabajo • Se usarán los módulos de pruebas JUnit propia del entorno de desarrollo.

4.7 Modelo de casos de uso

Las siguientes ilustraciones muestran el conjunto de casos de uso documentados para la realización del proyecto.

Como se observa en la Ilustración 6 se muestran las acciones que hace el sistema móvil, desde el despliegue de la bienvenida, la descarga de información en base al número telefónico capturado y su conexión con el cinturón materno fetal. Por su parte el usuario es quien ingresa el número telefónico para la descarga, por último, el sistema web le envía esta información al sistema móvil.

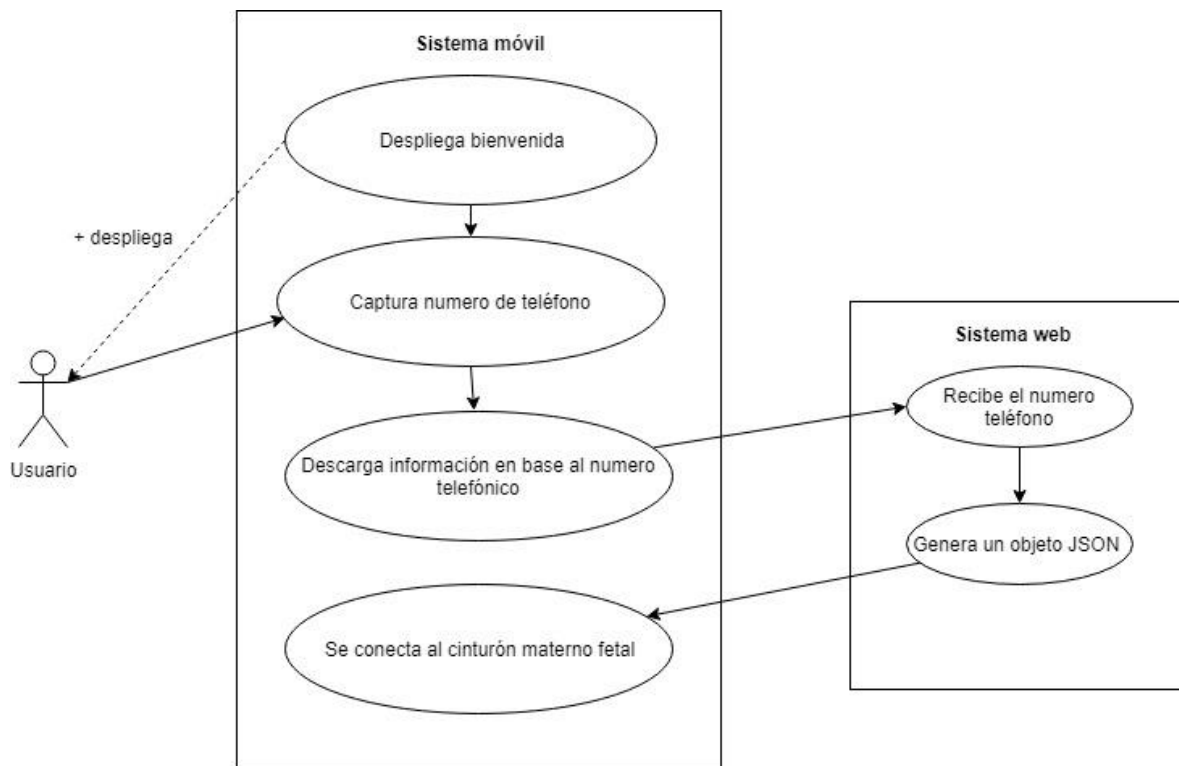


Ilustración 7 Modelo de casos de uso número 1.

El siguiente caso de uso Ilustración 7 muestra la funcionalidad de la obtención de datos del cinturón materno, como lo son las frecuencias cardíacas, la presión arterial y las alertas que este mismo puede generar, como vimos en los requerimientos pueden ser bajas o altas. Aquí interviene otro actor; el personal médico que es notificado en caso de que se genere alguna alerta, al igual que se notifica al sistema web.

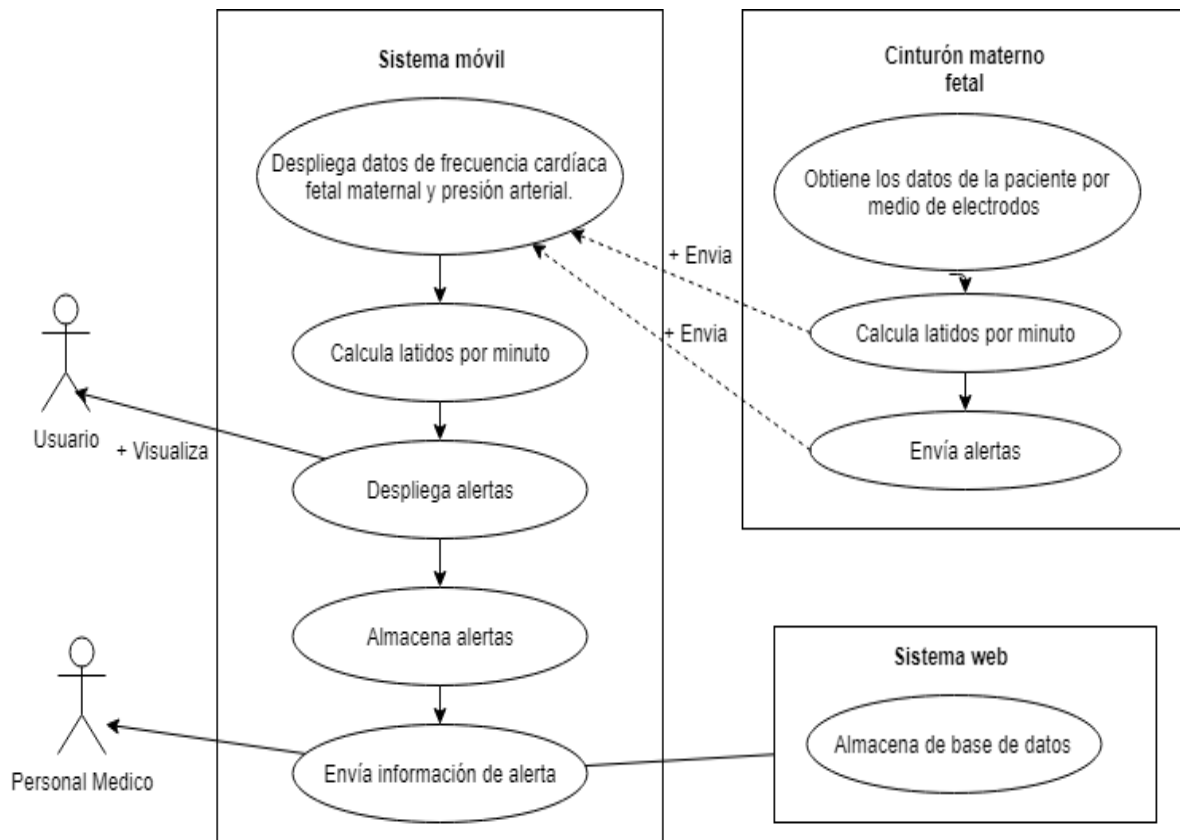


Ilustración 8 Modelo de casos de uso 2

EL ultimo diagrama de caso de uso Ilustración 8 hace referencia a las funcionalidades de los requerimientos donde se descarga una serie de información del sistema web para estar a la mano del usuario, las culturas próximas, los medicamentos activos y la información del doctor o del hospital, de nuevo intervienen dos usuarios, el usuario del sistema que consulta esta información desplegada por el sistema móvil y el personal médico que es alertado, en caso de que el usuario gente una alerta manual.

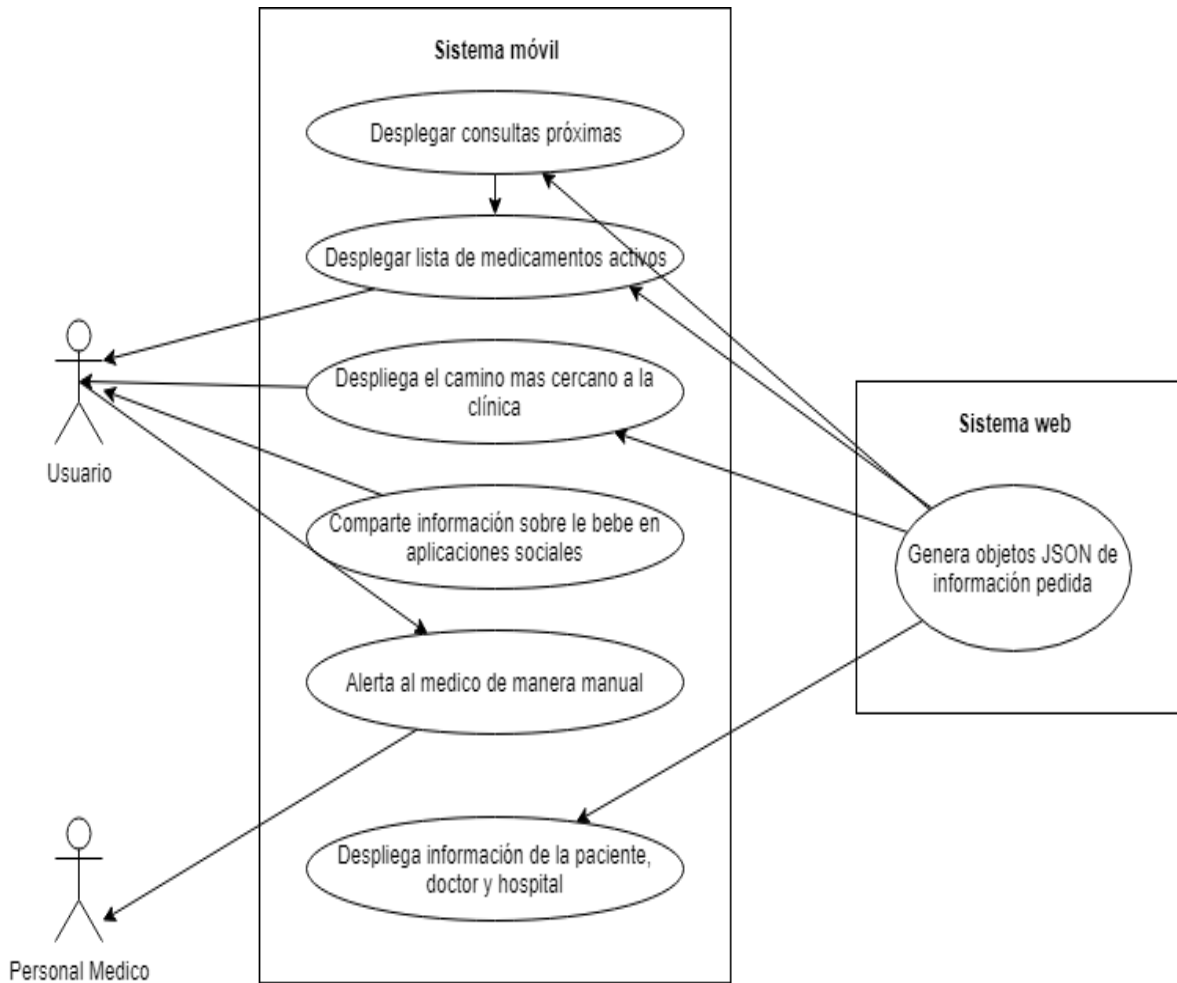


Ilustración 9 Modelo de casos de uso 3

4.8 Diagrama de secuencia

Una vez contemplado todos los componentes del sistema móvil se especifica en el diagrama de secuencia el funcionamiento de los principales eventos del sistema, reflejo de los casos de uso y los modelos de datos y clases, el diagrama representa los cambios que provocan los eventos atreves del tiempo.

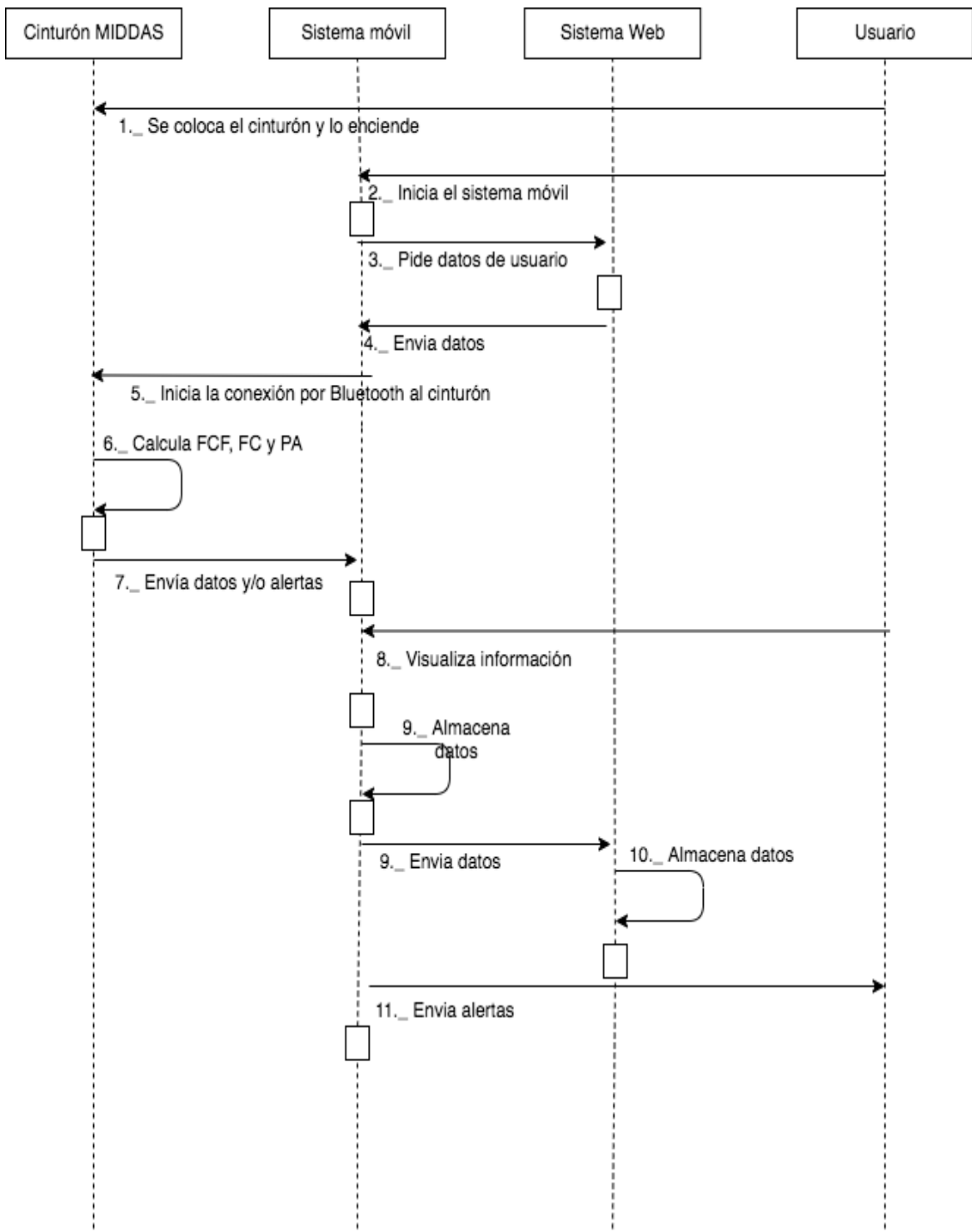


Ilustración 10 Diagrama de secuencia del sistema

4.9 Diseño del icono de aplicación

Se diseñó una imagen representativa del sistema para su fácil ubicación dentro de la interfaz del sistema operativo Android, el diseño fue libre, pero se basó en la funcionalidad principal que es medir los diferentes parámetros que genera el corazón latiendo de una persona.



Ilustración 11 Icono de aplicación

4.10 Diseño de interfaces

Las interfaces del usuario se describen en la siguiente sección, estas interfaces sirven para la aceptación y el diseño final de la aplicación.

4.10.1 Diseño de interfaz: pantalla de bienvenida

Esta pantalla recibe el inicio de la aplicación contara con una imagen de la empresa o del sistema un campo para introducir el número de teléfono de la embarazada y un botón para continuar.



Ilustración 12 Diseño de pantalla bienvenida.

4.10.2 Diseño de interfaz: conexión con el cinturón

Para la pantalla de conexión con el cinturón materno se diseñó una lista con los dispositivos con nombre y su identificador único, al momento de presionar un elemento de la lista se auto conectará.

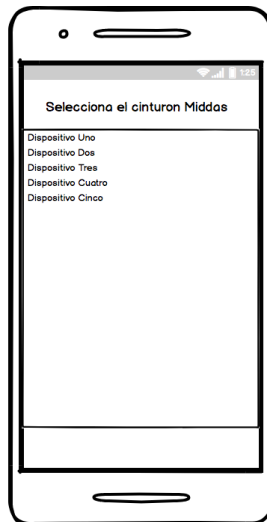


Ilustración 13 Diseño de pantalla conexión.

4.10.3 Diseño de interfaz; Visualización de datos

Para la visualización de datos se diseñaron tres vistas, la primera de ella mostrara en la parte superior la frecuencia cardiaca fetal y debajo de ella la frecuencia cardiaca de la madre, ambas con la leyenda de “latidos por minuto”.

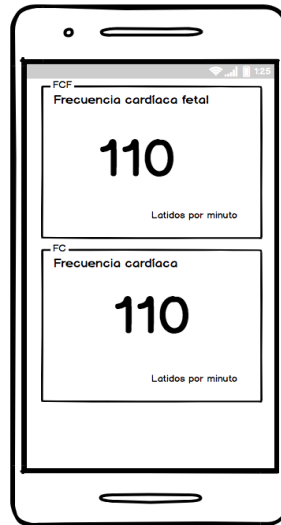


Ilustración 14 Diseño de pantalla frecuencias cardiacas.

La segunda pantalla que se creará para la visualización de datos comprenderá la presión arterial en una solo vista, con la presión sistólica en la parte superior y la diastólica en la inferior.

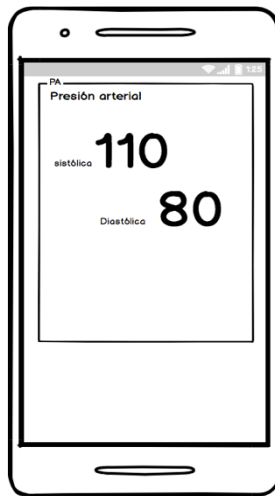


Ilustración 15 Diseño de pantalla presión arterial.

Por último, en la visualización de datos, se crea un calendario con la información de la presión arterial por día, desde el primer uso del cinturón. Cuando se presione sobre un día del mes, se despliega un pequeño menú con la información de la presión arterial sistólica y diastólica.



Ilustración 16 Diseño de pantalla calendario de presión arterial.

4.10.4 Diseño de interfaz: Información de paciente

Para la pantalla de información del paciente (nombre, apellido, edad, semanas de embarazo), se anexa también la información del médico y del hospital donde es atendida y recibió el servicio del cinturón materno feta.

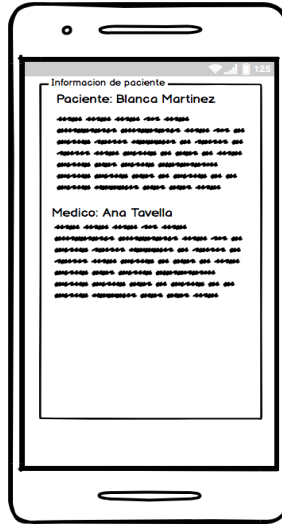


Ilustración 17 Diseño de pantalla información.

4.10.5 Diseño de interfaz: Información de medicamentos

Para esta pantalla se genera una lista medicamentos (nombre, aplicación, descripción y una imagen), solo se mostrarán los medicamentos que estén en medicación.

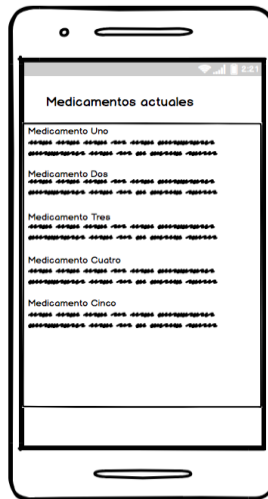


Ilustración 18 Diseño de pantalla medicamentos.

4.10.6 Diseño de interfaz: Consultas próximas

En la pantalla de consultas próximas se usa un diseño con un calendario y se resaltan con un color diferente los días donde la paciente tiene una cita médica, al presionar el día en el calendario se despliega una ventana con información de la cita.



Ilustración 19 Diseño de pantalla consulta.

4.10.7 Diseño de interfaz: Reacción a alertas

Las alertas del sistema visualmente serán una pantalla que puede emerger en cualquier actividad del sistema. Dara una pequeña nota donde informa que alertó al médico a cargo y preguntara si desea trasladarse de inmediato al hospital.

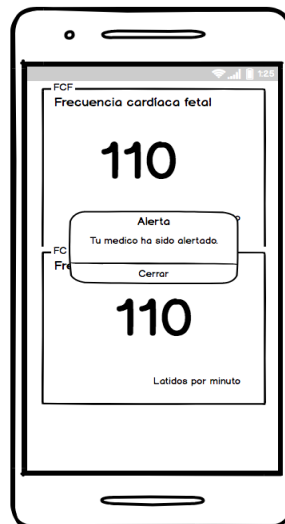


Ilustración 20 Diseño de pantalla de alertas.

4.11 Modelo de clases

Para el mapeo de información de los datos de usuario, del doctor, el hospital, las consultas próximas y los medicamentos se siguió la siguiente estructura del desarrollo de clases, cabe destacar que Android Studio maneja una clase por pantalla, sin embargo, los datos, y métodos ahí empleados no necesitan una descripción.

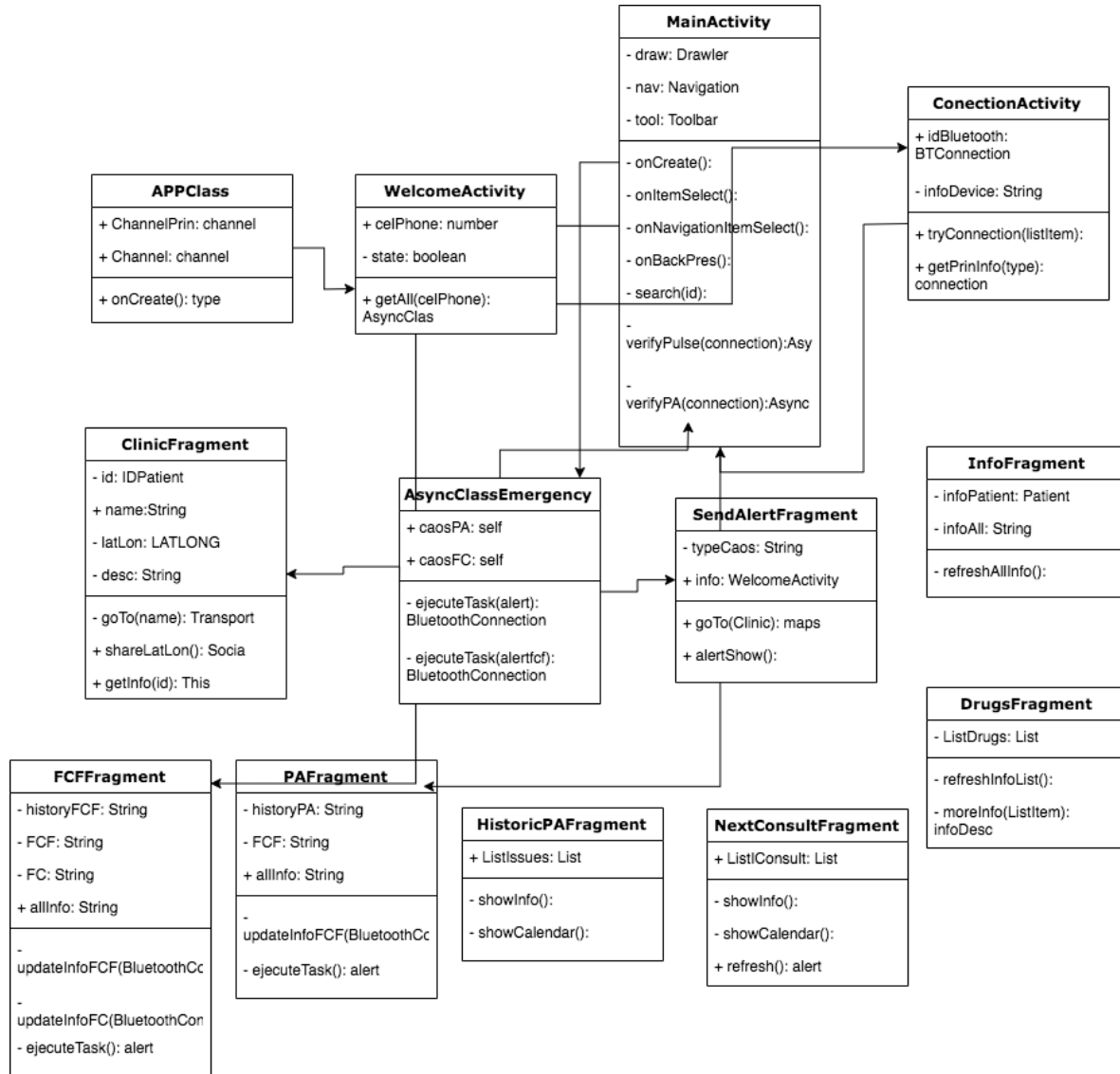


Ilustración 21 Modelo de clases.

4.12 Diagrama de datos

La clase para hacer uso de los datos del objeto JSON con la información y el software de SQLite para el manejo de la base de datos. Por características del sistema no es posible tener relaciones entre las tablas, pero esas funcionalidades se ajustan en el código de la aplicación.

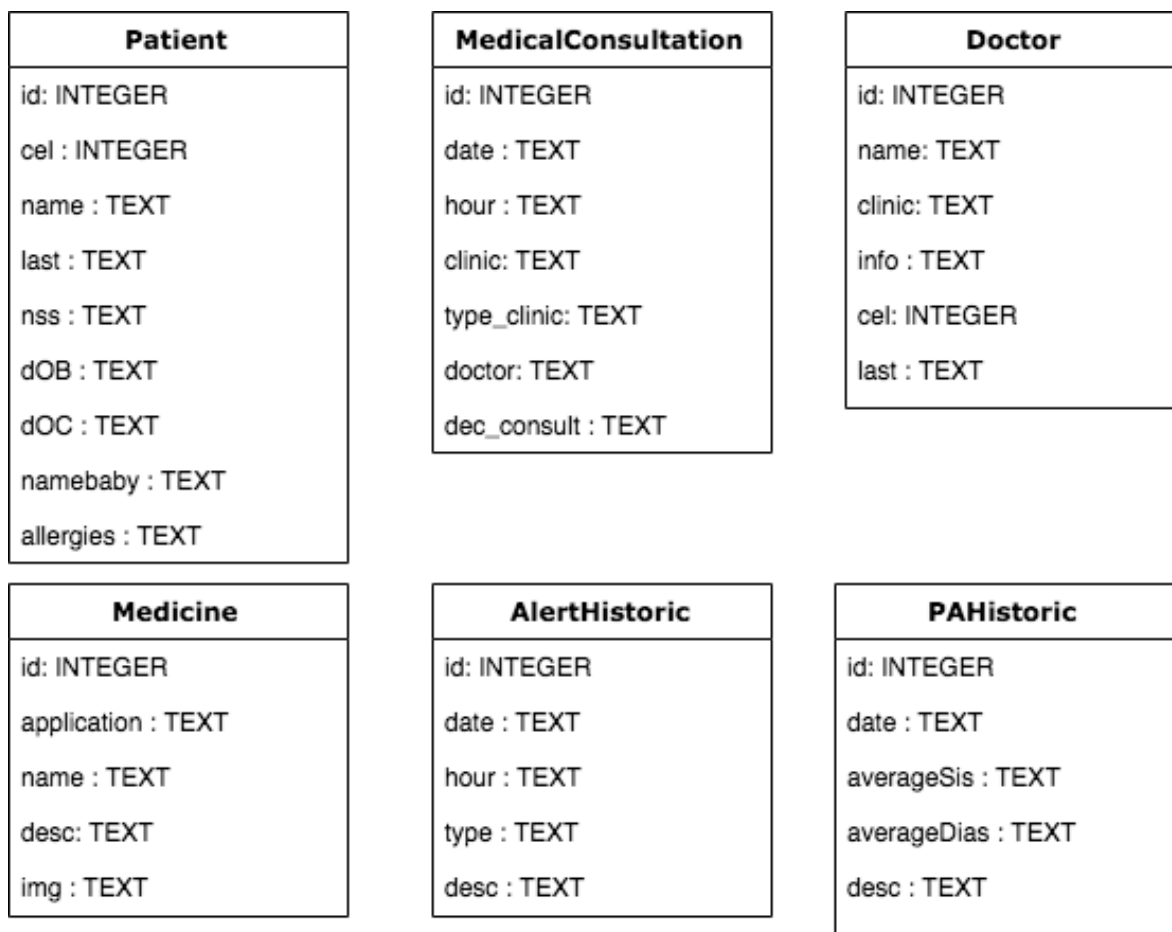


Ilustración 22 Diagrama de base de datos.

4.13 Codificación

El desarrollo del sistema se realizó en el entorno de desarrollo integrado oficial del sistema operativo Android. A continuación, se muestran los componentes y herramientas del entorno principales donde se programan las clases y recursos del sistema.

En la parte superior del explorador de archivos se encuentra la carpeta raíz del proyecto la cual almacena todos los archivos de la aplicación, el primer archivo rescatable del proyecto llamado `AndroidManifest.xml` es un archivo de configuración general de la aplicación es un archivo escrito en el lenguaje de etiquetas XML, en él está la información más importante del proyecto; el nombre, las versiones con las cuales es compatible, las actividades de cada clase y los permisos sobre el uso del hardware u otros requerimientos que necesita la aplicación para su correcto funcionamiento.

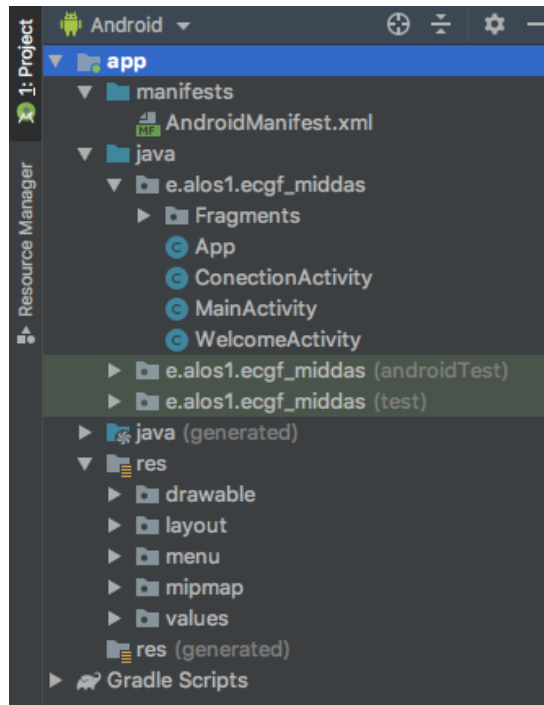


Ilustración 23 Ubicación de Clases.


```

package e.alos1.ecgf_middas.Fragments;

import androidx.appcompat.app.AppCompatActivity;

public class FCFragment extends Fragment {

    Context mContext;
    TextView idBufferIn;
    TextView txtfcf;
    private NotificationManagerCompat notificationManager;

    Handler bluetoothIn;
    int handlerState = 0;
    BluetoothAdapter btAdapter = null;
    BluetoothSocket btSocket = null;
    String bluetoothIn;
    StringBuilder dataStringIn = new StringBuilder();
    ConnectedThread myConnectonBT;
    //Identificador unico de servicio -SPP UUID
    UUID BT_UUID_UUID = UUID.fromString("00001101-0000-1000-8000-000057F9314E");
    // String para la direccion MA
    String address = null;
    DecimalFormat df = new DecimalFormat("#");

    private View rootView = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mContext = getActivity().getApplicationContext();

        bluetoothIn = (Handler) handleMessage(msg) = {
            if (msg.what == handlerState) {
                String readMessage = (String) msg.obj;
                dataStringIn.append(readMessage);

                int endOfLineIndex = dataStringIn.indexOf("\n");

                if (endOfLineIndex > 0) {
                    String dataInPrint = dataStringIn.substring(0, endOfLineIndex);
                    double result = Double.parseDouble(dataInPrint);
                    result = result + 20;
                    ejecutarAlertas(result);
                    txtfcf.setText(" " + df.format(result)); //bebe
                    result = result - 20;
                    Toast.makeText(mContext, " " + dataInPrint + " " + Toast.LENGTH_SHORT).show();
                    dataStringIn.delete(0, dataStringIn.length());
                }
            }
        };
    }
}

```

Ilustración 24 Ejemplo de clase FCFragment.

La programación principal del sistema se aloja en una carpeta llamada JAVA, dentro de ella una carpeta con el nombre del proyecto con las diferentes clases que necesita el sistema para funcionar, las clases están escritas en el lenguaje JAVA como se muestra en la ilustración 22 pero algunas toman clases del sistema escritas en C, C++ o Kotlin, dentro de este menú se accede a todos los recursos del proyecto, para la navegación entre pantallas se usó un recurso de diseño llamado NavigationDrawer el cual consta de una barra de navegación en la parte izquierda de la pantalla, la lógica de funcionamiento se programa con el uso de Clases llamadas Fragmentos los cuales actualizan las vistas del sistema dependiendo de la funcionalidad que se selecciona en la barra de navegación. Los fragmentos están escritos en Java y se encuentran en una carpeta llamada Fragments dentro de las clases de Java. Un poco más abajo en el menú de la ilustración 23 se encuentra otra carpeta llamada drawable en la cual se almacenan las imágenes o recursos gráficos que usara la aplicación, la carpeta layout por su parte guarda archivos tipo XML, que componen las vistas del sistema, todas las pantallas están escritas en este lenguaje de etiquetas, usualmente hay un archivo XML por tamaño de pantalla, siendo chico, mediano, grande y extra grande los patrones de diseño más recurrentes, un archivo muy importante que se encuentra en esta sección es strings.xml, este archivo es una lista de todos los recursos de textos usados en el sistema.

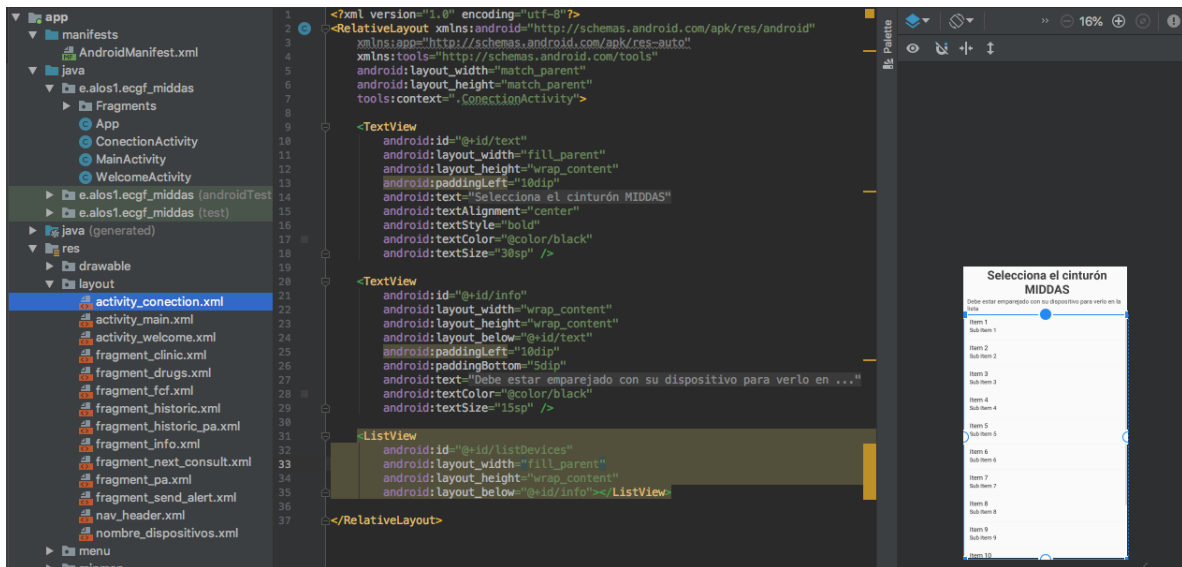


Ilustración 25 Ejemplo de archivo XML.

Por último, en la parte inferior se encuentra un archivo llamado Gradle Scripts que opera un sistema de automatización de construcción de código abierto el cual construye una gran variedad de complementos, plugins y librerías necesarias en el sistema sobre los conceptos de Apache Ant y Apache Maven, las librerías para la visualización de algunas características graficas como Material Desing se deben descargar desde este módulo.

Para la funcionalidad de la comunicación bluetooth con el cinturón materno fetal se desarrolló un dispositivo emulador del cinturón fetal Ilustración 25, certificado por la empresa, el objetivo principal de este dispositivo al igual que el cinturón, es obtener dos señales de tipo electrocardiograma, lograr una conexión por medio de Bluetooth, enviar alertas y las señales de ambos electrocardiogramas.

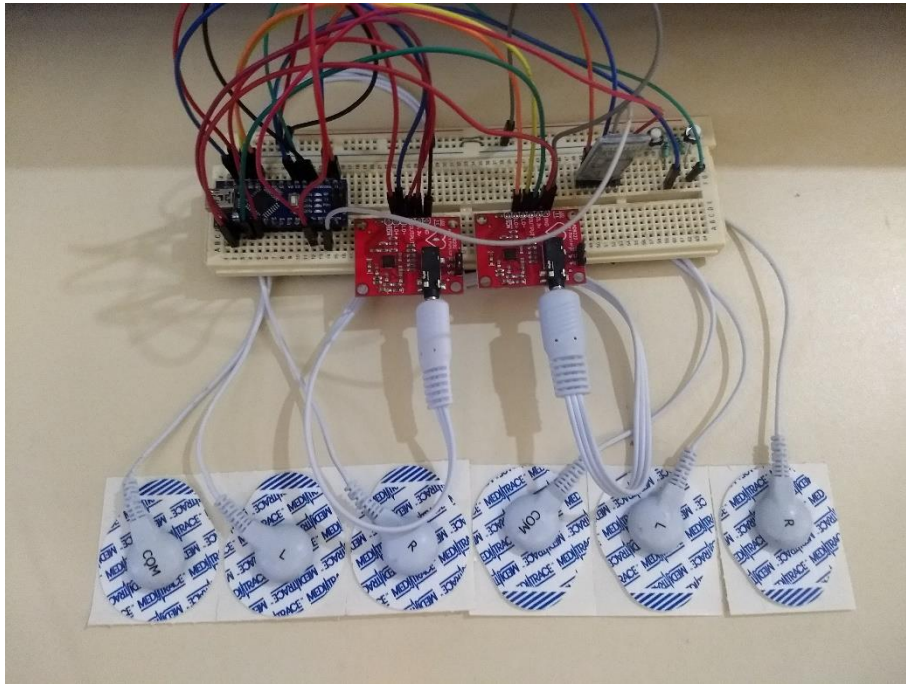
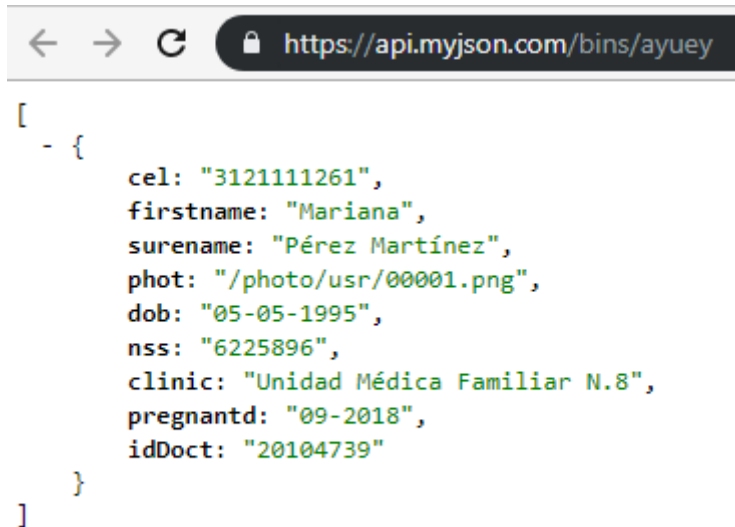


Ilustración 26 Dispositivo emulador del cinturón fetal.

Para el desarrollo del simulador del cinturón se usó una tarjeta Arduino, un módulo de comunicación Bluetooth, dos módulos AD8232 Heart Rate Monitor los cuales son circuitos que sirven para medir la actividad eléctrica del corazón, ambos dispositivos funcionan mediante electrodos colocados en la piel, se usan dos personas para usar señales diferentes, como lo haría un el ecocardiograma fetal y otro maternal, aunque no es rudimentariamente un dispositivo médico (ANALOG, 2018), el protocolo de comunicación de datos que envía es muy similar al que genera el cinturón de la empresa MIDDAS, también se incluyeron dos botones para simular alertas altas y bajas. Las alertas son simuladas ya que se necesitaría encontrar una persona con una frecuencia demasiado alta para poder disparar una alerta alta, o por su contrario una persona con un ritmo cardiaco muy bajo, la presión arterial se adjunta al flujo de datos que envía el dispositivo como una cadena de texto con los parámetros de presión arterial sistólica y diastólica.



```
[
  - {
    cel: "3121111261",
    firstname: "Mariana",
    surename: "Pérez Martínez",
    phot: "/photo/usr/00001.png",
    dob: "05-05-1995",
    nss: "6225896",
    clinic: "Unidad Médica Familiar N.8",
    pregnantd: "09-2018",
    idDoct: "20104739"
  }
]
```

Ilustración 27 Archivo JSON de datos básicos de paciente.

El sistema móvil se complementa de dos sistemas más, el cinturón fetal y el sistema de citas de MIDDAS, como ambos sistemas no pueden ser probados directamente con el sistema móvil, se utilizó un emulador del cinturón certificado por la empresa y se crearon archivos de tipo JSON con la información de la paciente.

4.14 Implantación en el proyecto

El sistema se terminó, en la siguiente sección se muestran las funcionalidades en la pantalla resultado del desarrollo del proyecto.



Ilustración 28 Pantalla de bienvenida.

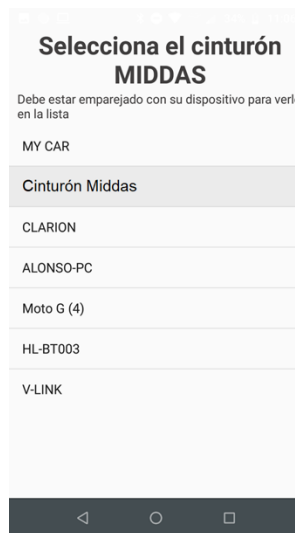


Ilustración 29 Pantalla de conexión.

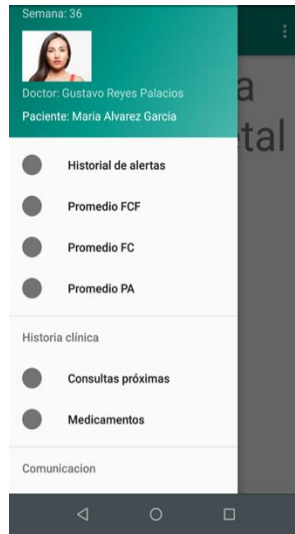


Ilustración 30 Pantalla de menú lateral.

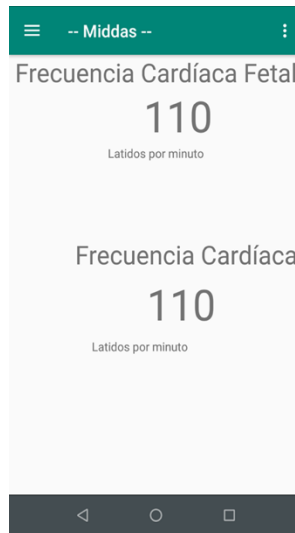
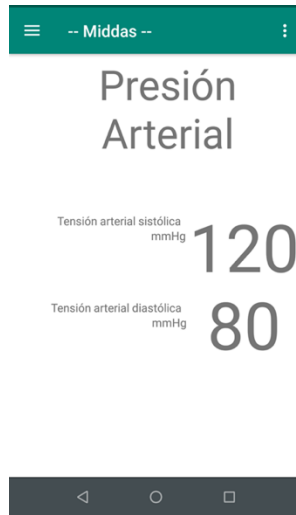
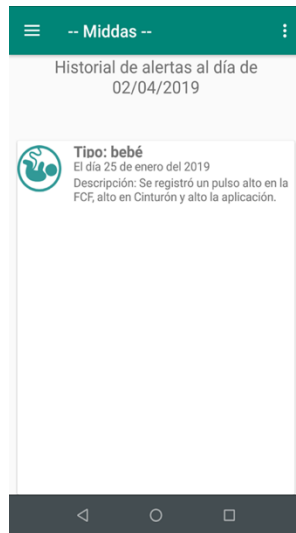


Ilustración 31 Pantalla de visualización de datos.



Ilustraci3n 32 Pantalla de visualizaci3n de datos (2).



Ilustraci3n 33 Pantalla de historial de alertas.

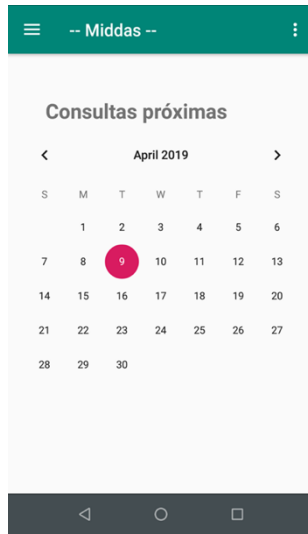


Ilustración 34 Pantalla de consultas próximas.

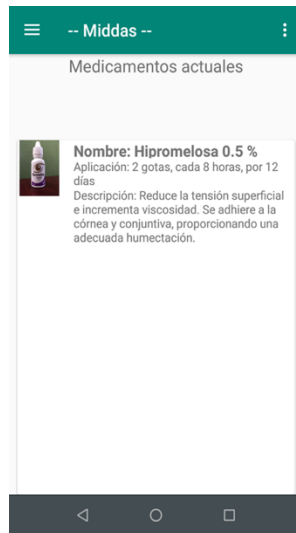


Ilustración 35 Pantalla de medicamentos activos.

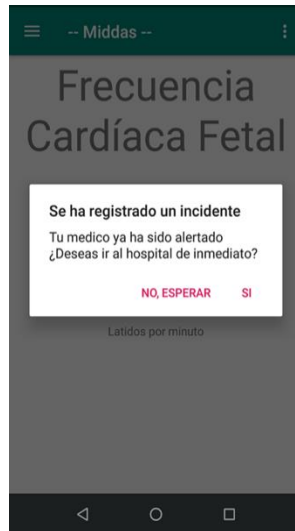


Ilustración 36 Pantalla de reacción ha alertas.

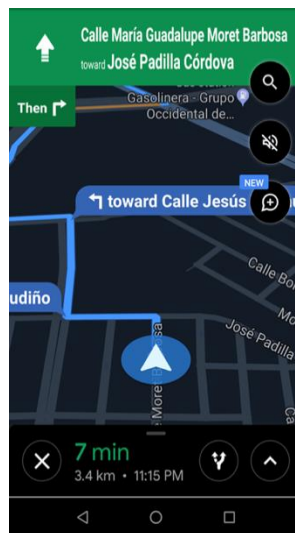


Ilustración 37 Pantalla de Google mapas.

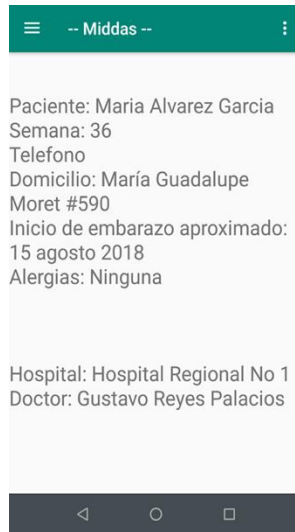


Ilustración 38 Pantalla de información de paciente.

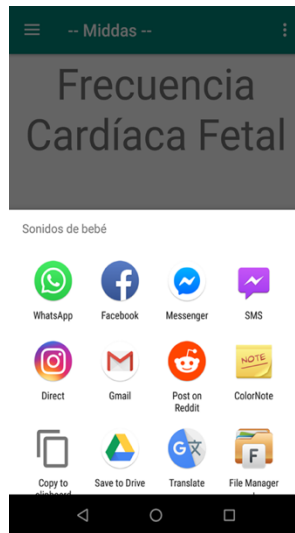


Ilustración 39 Pantalla de compartir estado.

4.15 Desarrollo de prototipos

Primer prototipo

Para el primer prototipo el sistema debería obtener la señal del cinturón de la empresa, calcular las frecuencias cardiacas, actuar cuando alguna de ellas saliera de un rango normal y tomar acciones en caso de que esto sucediese. Se consideró cargar por completo el cálculo de las frecuencias cardiacas al sistema móvil, sin embargo, se descartó de inmediato pues el flujo era tan rápido que ocasionaba un cuello de botella, resultado en una pérdida de datos importantes para calcular un promedio de pulsos cardiacos por minuto.

Segundo prototipo

Para el segundo prototipo se filtró una parte de los datos para evitar el exceso que estos generales al ser procesados y calculados por el sistema móvil, este filtro, trabaja del lado del sistema móvil, evitando valores ajenos al segmento de la onda "R" de un electrocardiograma.

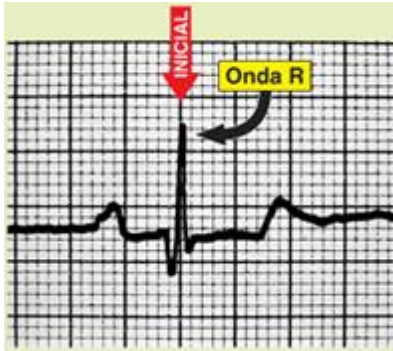


Ilustración 40 onda R (Ximena Rosana Rivera G., 2009)

Tercer prototipo

Este punto es donde se agrega una mayor cantidad de funcionalidades, en primera instancia el cinturón de la empresa, se adicionó un monitoreo de la presión arterial, también el filtrado de la señal como en la ilustración 40 se hace en el circuito del cinturón y no en el sistema móvil, por ende, las alertas también son enviadas desde el cinturón, las respuestas a estas siguen en el sistema móvil.

Cuarto prototipo

Este entregable del prototipo consume un web servicio del sistema de pacientes de la empresa y carga algunos datos referentes al monitoreo durante el embarazo, también se modificaron las interfaces del sistema para darle un mejor acabado visual.

Quinto prototipo

El ultimo prototipo era totalmente funcional, se hicieron pruebas de funcionalidad con un simulador del cinturón fetal con comunicación bluetooth, logando una buena respuesta en caso de riesgo fetal, el consumo de web servicio también se probó obteniendo resultados satisfactorios.

4.16 Pruebas

Las pruebas se desarrollaron para las tres funcionalidades principales, el consumo de servicios web para obtención de información de la paciente, las pruebas de conectividad Bluetooth entre el cinturón y el sistema móvil y las pruebas en caso de emergencias, como resultado se redujo el código de seguimiento de alertas para optimizar la respuesta ante una emergencia.

Capítulo 5

Conclusiones y recomendaciones

5.1 Introducción

En el presente capítulo se presentan las conclusiones del desarrollo de la presente tesis, además de las recomendaciones para trabajos futuros.

5.2 Resultados

En base a los resultados obtenidos de este trabajo de investigación se comprobó que el uso de un sistema móvil resulta ser una herramienta eficiente para el monitoreo de parámetros durante el embarazo y a su vez una alternativa rápida de respuesta a problemas que detonan de alteraciones la frecuencia cardiaca fetal y la presión arterial durante el embarazo.

El principal motivo de su efectividad es que mitiga el uso de recursos de monitoreo del cinturón materno, para procesarlos en el sistema móvil del celular de la paciente, formando un sistema modular que predice de una manera más eficiente las complicaciones que pudiera llegar a tener la usuaria, y finalmente alertando de manera oportuna al personal médico.

5.3 Conclusiones y trabajo futuro

Se puede concluir que el uso de sistemas móviles o aplicaciones en Smartphones pueden ser una gran ventaja para mitigar y repartir bien el procesamiento en sistemas embebidos en el área de la salud.

Como trabajo futuro se propone el desarrollo en el sistema operativo IOS, para abarcar un mayor mercado grande de Smartphone.

También puede ser la integración de un sistema con técnicas de minería de datos para predecir posibles complicaciones cardiovasculares, con datos obtenidos del cinturón y enviados a través del sistema móvil.

Bibliografía

- Akshada, M., & Patil, R. (2018). Fetus Heart Rate Monitoring Using Smart Phone. 7(3), 757-761.
- ALONSO, F., MARTINEZ, L., & SEGOVIA, F. J. (2005). , Introducción a la Ingeniería del software Modelos de desarrollo de programas.
- ANALOG, D. (2018). Single-Lead , Heart Rate Monitor Front End Datasheet. 1-28.
- Astivia, D., Gayt, D., & Luna, O. (2014). Electrocardiógrafo materno-fetal abdominal: Prototipo portátil . 176-179.
- Báez, M. (2012). Introducción - ¿ Qué Es Android ?
- Baz, A., Ferreira, I., Álvarez, M., & García, R. (2011). Dispositivos móviles. 1-12. Retrieved from <http://isa.uniovi.es/docencia/SIGC/pdf/telefoniamovil.pdf>
- Bellabeat. (2016). Introduces Free App and Add-On that Safely Detect Fetal Heartbeat. Retrieved from https://www.bellabeat.com/assets/img/press/Shell_by_Bellabeat_November_2016.pdf
- Bluetooth. (n.d.). Retrieved from <https://developer.android.com/guide/topics/connectivity/bluetooth?hl=es-419>
- Caballero, F., Jiménez, V., Sánchez, A., & López, A. (2017). Prototipo De Electrocardiógrafo Portátil. 39(04), 82-99.
- Dalia, P., & González, P. (2018). Diseñan sistema para detectar sufrimiento fetal. 2018.
- Das, A., Pradhapan, P., Groenendaal, W., Adiraju, P., Rajan, R. T., Catthoor, F., ... Van Hoof, C. (2018). Unsupervised heart-rate estimation in wearables with Liquid states and a probabilistic readout. *Neural Networks*, 99(August), 134-147. <https://doi.org/10.1016/j.neunet.2017.12.015>
- De La Cruz, B., Cuellar, R., Rojas, E., Molina, V., & Robles, H. V. (2015). Transmission of ECG signals with android mobile system via bluetooth. *Pan American Health Care Exchanges, PAHCE*, 2015-July. <https://doi.org/10.1109/PAHCE.2015.7173316>
- Dueñas-García, O. F., & Díaz-Sotomayor, M. (2011). Controversias e historia del monitoreo cardiaco fetal. *Revista de Investigacion Clinica*, 63(6), 659-663.
- Educaci, D. E., Box, C. P. O., Rico, C. P., Prof, R., Cabrera, L., & Reservados, D. (2015). *Lectura-Comprension-EKG*.
-

Funciones de Android Studio. (n.d.). Retrieved from <https://developer.android.com/studio/features>

Geografía, I. N. de E. y. (2017). Estadísticas Vitales. Defunciones Fetales 2017. Retrieved from http://www.inegi.org.mx/sistemas/olap/consulta/general_ver4/MDXQueryDatos.asp?#Regreso&c=11143

Grado, T. D. E., Manuel, A., & Vera, A. (2015). " Optimización de la plataforma para el registro y visualización de ECG en el Sistema Operativo Android: Almacenar y Visualizar ECG cargados . ."

Health, S. C. (n.d.). Control externo e interno de la frecuencia cardíaca fetal. Retrieved December 22, 2019, from Stanford Health website: <https://www.stanfordchildrens.org/es/topic/default?id=control-externo-e-interno-de-la-frecuencia-cardaca-fetal-92-P09290>

Maldonado, S.-, Angel, A., Elena, M., Cano-anguiano, J. E., & Guadalupe, R. De. (2018). Revisión literaria tecnológica de dispositivos y métodos para monitoreo de la frecuencia cardiaca fetal Literary technological review of devices and methods for monitoring of fetal heart rate Revista de Prototipos Technol. Revista de Prototipos Tecnológicos, 4(14), 23–29.

OBS Business School. (2019). ¿Qué son las metodologías de desarrollo de software? Retrieved August 12, 2019, from <https://obsbusiness.school/int/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software>

por el Bachiller, P., Robertho Luty Artica Navarro Asesor, B., & Luis Honorato Pita Astengo, I. (2014). DESARROLLO DE APLICACIONES MÓVILES. Retrieved from http://repositorio.unapiquitos.edu.pe/bitstream/handle/UNAP/4515/Robertho_Tesis_Titulo_2014.pdf?sequence=1&isAllowed=y

PRESSMAN, R. (2010). INGENIERIA DEL SOFTWARE. UN ENFOQUE PRACTICO (Septima ed). McGraw-Hill Interamericana de España S.L.

R. Marín Iranzo, M. Gorostidi Pérez, R. Á.-N. (2012). Hipertensión arterial y embarazo. (August), 1–15. <https://doi.org/10.3265/NefroPlus.pre2011.Jun.10997>

Radha, B. K. (2017). Real -Time Ambulatory Monitoring System. International Research Journal of Engineering and Technology(IRJET), 4(5), 605–611. Retrieved from <https://www.irjet.net/archives/V4/i5/IRJET-V4I5119.pdf>

RANGEL BREÑA, J. C., ARCINIEGA MONTIEL, S., & SANTIAGO RAMIREZ, J. J. (2017). MÉTODO PARA LA ADQUISICIÓN, FILTRADO Y CARACTERIZACIÓN DE LA SEÑAL CARDIACA FETAL. Mexico.

Robledo Fernández, D., & Robledo, D. (2014). Desarrollo De Aplicaciones Para Android. 49. Retrieved from https://books.google.com.ec/books?id=IwLXAAwAAQBAJ&pg=PT135&dq=desarrollo+de+aplicaciones++para+android+II&hl=es&sa=X&redir_esc=y#v=onepage&q=desarrollo de aplicaciones para android II&f=false

Sarmiento, L. O., Vélez, I., Flórez, N., Otero, J. C., Manrique, F., & Millet, J. (2010a). Estimación no invasiva de la frecuencia cardiaca fetal mediante técnicas de separación ciega de fuentes. *Revista Colombiana de Cardiología*, 17(4), 187–190. [https://doi.org/10.1016/S0120-5633\(10\)70238-2](https://doi.org/10.1016/S0120-5633(10)70238-2)

Sarmiento, L. O., Vélez, I., Flórez, N., Otero, J. C., Manrique, F., & Millet, J. (2010b). Estimación no invasiva de la frecuencia cardiaca fetal mediante técnicas de separación ciega de fuentes. *Revista Colombiana de Cardiología*, 17(4), 187–190. [https://doi.org/10.1016/S0120-5633\(10\)70238-2](https://doi.org/10.1016/S0120-5633(10)70238-2)

Sommerville, I. (2005). INGENIERIA DEL SOFTWARE. Séptima edición. <https://doi.org/http://zeus.inf.ucv.cl/~bcrawford/Modelado%20UML/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf>

Valencia, I. P. (2018). Método para caracterizar con mayor precisión señal cardiaca fetal. Retrieved from Agencia Informativa Conacyt website: <http://www.conacytprensa.mx/index.php/ciencia/salud/20585-metodo-mayor-precision-senal-cardiaca-fetal>

Ximena Rosana Rivera G. (2009). *Electrocardiografía Básica*.

Yuan, L., Yuan, Y., Zhou, Z., Bai, Y., & Wu, S. (2019). A Fetal ECG Monitoring System Based on the Android Smartphone. *Sensors* (Basel, Switzerland), 19(3). <https://doi.org/10.3390/s19030446>