



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“SISTEMA DE MONITORIZACIÓN CONTINUO DEL RITMO
CARDÍACO CON DISPOSITIVO PORTÁTIL PARA PACIENTES
CON PROBLEMAS DEL CORAZÓN”

INFORME DE MATERIA INTEGRADORA

Previo a la obtención del Título de:

INGENIERO EN TELEMÁTICA

BYRON VICENTE NARVÁEZ RIVADENEIRA

MARTHA RAQUEL VILLACÍS DÁVILA

GUAYAQUIL – ECUADOR

AÑO: 2016

AGRADECIMIENTOS

Agradezco primeramente a nuestro Padre Celestial, quien es lo más importante en mi vida, y quien me ha dado la sabiduría y las fuerzas para cumplir con este objetivo, a mis padres, quienes con su esfuerzo me han demostrado lo que es la superación. A los maestros, que me brindaron sus conocimientos al momento de aconsejar e impartir sus enseñanzas, un especial agradecimiento al ingeniero Washington Velásquez, con quien compartí estos últimos años de mi carrera, a mi tutora de tesis la ingeniera Marjorie Chalén, de quien recibí gran ayuda y consejos. Y de manera muy afectiva y personal agradezco a todos mis compañeros del club de robótica “Robota”, por haber compartido conmigo tantas experiencias dentro y fuera del país, en especial a mis grandes amigos George, Edgar, Fray, Saúl, José, Luis, Joselyn, Tamara, Gema y Guillermo por brindarme su apoyo y amistad.

Byron Vicente Narváez Rivadeneira

Mis más sinceros agradecimientos a Dios por permitirnos llegar a culminar nuestra carrera universitaria, por sus innumerables bendiciones, a mis padres que con mucho esfuerzo y dedicación nos formaron día a día para dar el mayor potencial. Agradecemos a nuestros tutores de proyecto, a los ingenieros: Washington Velásquez Vargas y Marjorie Alexandra Chalén Troya quienes nos dieron su apoyo y guía a lo largo del desarrollo del proyecto, al Ing. Abraham Morán que siempre estuvo pendiente y predispuesto en colaborar con nosotros.

Martha Raquel Villacís Dávila

DEDICATORIA

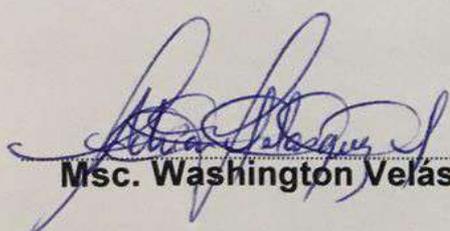
El presente proyecto lo dedico a mis padres Miguel Narváez y Victoria Rivadeneira, que siempre han estado apoyándome sin condición y en todo momento, en especial a mi Madre, quien ha sido la persona que con la ayuda de Dios ha mantenido esta familia unida y en los caminos de Dios. A mi padre, que ha sido el sostén de la familia, que con su incomparable empeño ha demostrado lo que es la fuerza y el carácter para abordar cualquier responsabilidad. Y por último a mis hermanos Víctor y Pablo con quienes siempre compartimos el sueño de salir adelante.

Byron Vicente Narváez Rivadeneira

Le dedico este proyecto a Dios y a mis padres Hugo Alfredo Villacís y a mi madre Martha Dávila García por estar a mi lado en estos años de estudio, por su amor incondicional, dedicación, esfuerzo por enseñarme buenos valores y guiarme en el transcurso de mi vida, también a mis hermanos que siempre han estado apoyándome y brindaron ayuda. A mis docentes por brindarme sus conocimientos y contribuir con mi formación profesional. Finalmente, a mis amigos por apoyarme.

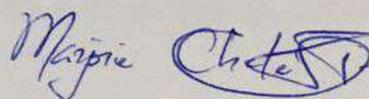
Martha Raquel Villacís Dávila

TRIBUNAL DE EVALUACIÓN



Msc. Washington Velásquez

PROFESOR EVALUADOR

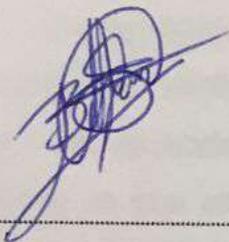


Msig. Marjorie Chalén

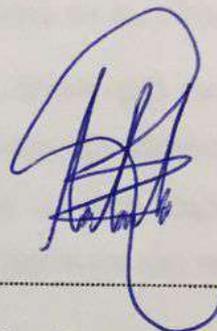
PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Byron Vicente Narváez Rivadeneira



Martha Raquel Villacís Dávila

RESUMEN

Las enfermedades cardiovasculares han ido en aumento con el pasar de los años con cifras alarmantes de fallecimientos o defunciones en el Ecuador, entre las principales razones: la mala alimentación, sedentarismo, estrés y en algunos casos se dan de forma hereditaria, algunas de estas enfermedades se dan de manera silenciosa con el pasar de los años, el paciente no se percata de los síntomas, por lo que los médicos especialistas recomiendan hacerse chequeos rutinarios de su ritmo cardíaco.

En el mercado existen productos con diseño de pulsera que brindan servicios de control del ritmo cardíaco por medio de los BPM (Latidos por Minuto), pero no cuentan con la comunicación paciente – especialista, por lo que representa una gran desventaja ya que estos datos solo quedan con el paciente sin ningún monitoreo o análisis de un médico conocedor del tema.

El presente trabajo plantea un sistema de monitoreo del ritmo cardíaco, el cual relaciona salud y tecnología, de fácil uso para médicos y pacientes que padezcan enfermedades referentes al corazón. Este sistema permite que cada paciente disponga de un dispositivo portable y de fácil manejo, que se encarga de realizar la lectura del ritmo cardíaco, un prototipo en forma de pulsera compuesta a nivel de hardware con los siguientes elementos: Sensor de pulso, Arduino Pro Mini, Módulo Sim 808 con sus respectivas antenas GSM / GPRS y GPS, con una batería LiPo de 3.7 V.

Con la ayuda de la tecnología 2G, específicamente GPRS, permite transmitir datos referentes al ritmo cardíaco, obtenidos por medio del sensor de pulso, como también las coordenadas de la ubicación geográfica del paciente, desde cualquier parte del mundo donde haya acceso a internet, estos datos pueden ser visualizados a través del aplicativo web, que cuenta con tres tipos de usuario: Paciente, Doctor y Administrador con diferentes permisos, que se comunican a través de mensajes internos o mails, el sistema permite ingresar el historial clínico de cada paciente todo en la aplicación web. Además, existen diferentes alertas que serán enviadas por mensajes de texto (SMS) y correos electrónicos al doctor encargado y al representante con el fin de alertar alguna anomalía en el ritmo cardíaco.

ÍNDICE GENERAL

AGRADECIMIENTOS	ii
DEDICATORIA	iii
TRIBUNAL DE EVALUACIÓN	iv
DECLARACIÓN EXPRESA	v
RESUMEN	vi
ÍNDICE GENERAL	vii
ÍNDICE DE FIGURAS	x
ÍNDICE DE TABLAS	xiii
CAPÍTULO 1	1
1. GENERALIDADES.	1
1.1 Antecedentes.	1
1.2 Justificación.....	4
1.3 Objetivos.....	5
1.3.1 Objetivo general.....	5
1.3.2 Objetivos específicos.....	5
1.4 Alcance y limitaciones.....	6
CAPÍTULO 2.....	7
2. MARCO TEÓRICO.	7
2.1 Componentes electrónicos.....	7
2.1.1 Sensor de pulso: Pulse Sensor.....	7
2.1.2 Módulo Sim808.....	9
2.1.3 Arduino Pro mini.....	11
2.1.4 Batería LiPo.....	13
2.2 Tecnologías de comunicación.....	14
2.2.1 Antena Gsm/Gprs.....	14
2.2.2 Antena Gps.....	15
2.3 Tecnologías de desarrollo.....	16
2.3.1 IDE de Arduino 1.6.11.....	17

2.3.2	Django.	18
2.3.3	Python.....	18
2.3.4	Html.....	19
2.3.5	Apache.	20
2.3.6	MySQL.....	20
2.4	Software complementario.	21
2.4.1	Autodesk Inventor Professional.....	21
2.4.2	Lucidchart.....	22
3.	IMPLEMENTACIÓN Y DESARROLLO.	23
3.1	Paciente.....	23
3.2	Dispositivo.	24
3.2.1	Diseño de la carcasa.....	24
3.2.2	Electrónica del dispositivo.....	26
3.3	Servidor web.	38
3.3.1	Adquisición de datos.	39
3.3.2	Modelo relacional de la base de datos.....	40
3.4	Aplicación web.	40
3.4.1	Posicionamiento y Google Maps.....	41
3.4.2	Gráficas en tiempo real.....	42
3.4.3	Envío de alertas.	43
3.5	Usuarios de la aplicación web.....	46
CAPÍTULO 4.....		47
4.	ANÁLISIS DE RESULTADOS.....	47
4.1	Análisis técnico.....	47
4.1.1	Resultados del Dispositivo.	48
4.1.2	Resultados de la aplicación web.	53
4.2	Análisis de costos.	59
4.2.1	Costo de elaboración del dispositivo.....	59
4.2.2	Costo del servidor web.	60

CONCLUSIONES Y RECOMENDACIONES.....	61
BIBLIOGRAFÍA	63
ANEXOS	68
ANEXO A – GLOSARIO.	68
ANEXO B – DISEÑO DE LA CARCASA EN 3D.....	70
ANEXO C – CODIGO DEL DISPOSITIVO EN ARDUINO.	72

ÍNDICE DE FIGURAS

Figura 2.1: Sensor de pulso cardíaco, “Pulse Sensor”, utilizado para realizar la lectura del ritmo cardíaco mediante un led y un fotosensor.....	8
Figura 2.2: Gráfica del electrocardiograma (ECG). [11].....	8
Figura 2.3: Modelo de gráfica de una fotopleletismografía (PPG). [13]	9
Figura 2.4: Módulo de comunicación Sim808, elaborado por la compañía SimCom, empleado para realizar el envío de datos del ritmo cardíaco, vía GPRS hacia la aplicación web.....	11
Figura 2.5: Placa de Desarrollo Arduino Pro-mini, utilizado para el procesamiento de la señal tomada por el sensor y luego enviarla hacia el módulo de comunicación.	13
Figura 2.6: Batería de Litio Polímero, de 3.7V y 2000mAh, utilizada para energizar cada uno de los elementos electrónicos del dispositivo.	14
Figura 2.7: Antena GSM/GPRS con conector ufl, empleada para aumentar la intensidad en la señal de comunicación GPRS.	15
Figura 2.8: Antena GPS Pasiva con conector ufl, utilizada para obtener las coordenadas de la posición geográfica del dispositivo.	16
Figura 2.9: Plataforma de desarrollo arduino.	17
Figura 2.10: Consola interactiva de Python.	19
Figura 2.11: Área de diseño en Autodesk Inventor Professional.	21
Figura 2.12: Área de trabajo de Lucidchart, empleado para los diagramas de flujo.	22
Figura 3.1: Diagrama de bloques general del sistema.....	23
Figura 3.2: Diseño de la carcasa en vista lateral.	24
Figura 3.3: Diseño de la base de la carcasa.....	25
Figura 3.4: Diseño del cobertor de la carcasa.	26
Figura 3.5: Diagrama esquemático electrónico del dispositivo.....	27
Figura 3.6: Asignación y configuración de pines digitales y analógicos usados en el módulo Arduino.	28
Figura 3.7: Lectura y almacenamiento de las variables PICO y VALLE que forman la fotopleletismografía.	30

Figura 3.8: Código de encendido del led de pulso cada vez que el sensor detecta un latido.	31
Figura 3.9: Código que calcula el tiempo que existe entre dos latidos y se almacena en la variable IBI.	31
Figura 3.10: Código que calcula el valor de la cantidad de latidos por minuto y la almacena en la variable BPM.	32
Figura 3.11: Código que calcula el valor de la amplitud y el umbral de la señal PPG.	33
Figura 3.12: Código que reinicia las variables cada vez que no se detecta algún latido en un tiempo mayor a 2,5 segundos.	33
Figura 3.13: Envío de datos hacia la función SubmitHttpRequest.	34
Figura 3.14: Función power, que enciende el módulo Sim808.	35
Figura 3.15: Configuración del módulo Sim808.	36
Figura 3.16: Función GPS, que realiza la búsqueda de las coordenadas por parte de la antena.	37
Figura 3.17: Código que realiza el envío de todos los valores hacia el servidor web.	38
Figura 3.18: Servidor gratuito “Python Any Where” donde se configura por medio de la interfaz gráfica en la pestaña web.	39
Figura 3.19: Modelo Relacional de la base de datos en MySQL.	40
Figura 3.20: Arquitectura de desarrollo web.	41
Figura 3.21: Función de posicionamiento de las coordenadas sobre el mapa de Google.	42
Figura 3.22: Función que dibuja los puntos de la fotopleitismografía.	43
Figura 3.23: Rango de alarmas para notificar al médico y al representante del paciente ante cualquier anomalía.	44
Figura 3.24: Archivo de configuración setting.py para realizar el envío de Correos Electrónicos.	44
Figura 3.25: Código de la función que realiza el envío de Correos electrónicos.	45
Figura 3.26: Código de la función que realiza el envío de mensajes de texto.	46

Figura 4.1: Diagrama de flujo de resultados del sistema.....	47
Figura 4.2: Prototipo final de la carcasa.	53
Figura 4.3: Página de registro de médicos.	54
Figura 4.4: Página de registro de Pacientes.	54
Figura 4.5: Página correspondiente a la interfaz de usuario del médico encargado.	55
Figura 4.6: Página que muestra la lista de pacientes según el médico que haya iniciado sesión.....	55
Figura 4.7: Página que muestra las principales alertas ya sea de peligro, advertencia, si esta en alguna actividad física o si ha llegado un mensaje nuevo.....	56
Figura 4.8: Página en donde se visualiza el mapa con la ubicación actual del paciente.....	57
Figura 4.9: Página en donde se visualiza la señal PPG (Fotopleetismógrafo) en tiempo real.	58
Figura 4.10: Envío de correo electrónico al detectar una alerta del sistema.	58

ÍNDICE DE TABLAS

Tabla 1: Tabla comparativa de costos de los diferentes módulos de comunicación celular.	11
Tabla 2: Cantidad de MB por períodos de tiempo en envíos de datos al servidor web.	51
Tabla 3: Costos en la implementación del dispositivo.	59

CAPÍTULO 1

1. GENERALIDADES.

En el presente capítulo se narra los antecedentes referentes a la problemática a la que se busca implementar una solución, también se justifica el desarrollo del proyecto, se detallan, el objetivo general y los objetivos específicos. De igual manera se describen los alcances y limitaciones del proyecto.

1.1 Antecedentes.

A través de los años la humanidad ha experimentado una gravísima cantidad de enfermedades, ubicándose entre las primeras razones de mortalidad las cardiovasculares con cifras alarmantes de fallecimientos entre hombres y mujeres por consecuencia de la mala alimentación, falta de ejercicios, estrés, consumo de alcohol, tabaquismo, causando una tasa del 30% de mortalidad global según la Organización Mundial de la Salud (OMS) y antes de estudios relacionados, entre las más comunes; arteriosclerosis, trombosis cardiopatía coronaria, hipertensión arterial, cardiopatía isquémica, arritmias y otras anomalías del músculo cardíaco. [1]

Entre las enfermedades silenciosas se encuentra la hipertensión arterial; considerada el “asesino silencioso”, que afecta directamente al corazón y al cerebro, dado que en la mayoría de los casos no presenta síntomas, esto se debe al aumento de la presión de forma gradual. Con el transcurrir del tiempo la persona se va adaptando a manejar niveles de presión elevados hasta producir por alguna eventualidad un infarto. [2]

Gran porcentaje de alteraciones del ritmo cardíaco se da de forma hereditaria, por lo cual muchas personas quedan expuestas a padecer alguna de ellas, por lo tanto, los médicos especialistas en cardiología, recomiendan realizar controles periódicos, para indicar la existencia de alguna anomalía. [3]

El ámbito del cuidado de salud a través de dispositivos móviles integra las tecnologías de computación e internet, el uso de sensores y los sistemas de comunicación e información. Se han implementado sistemas de telemedicina

compatible con las redes de telecomunicaciones móviles, que permiten a un médico controlar remotamente un paciente en libre movimiento o en situaciones de emergencia. [4]

En estos días existe una brecha muy corta entre los mundos de la física y la información gracias al uso de sensores, los cuales recogen datos del entorno, proporcionando información necesaria, la miniaturización y la nanotecnología dentro de las cosas o también conocidos mayormente como dispositivos inteligentes. La inteligencia integrada en un dispositivo permite que el mismo funcione de manera automática, pero con el fin de trabajar e interactuar con otros dispositivos, para lograr esto se requiere conectividad. La conexión que utilizan los equipos es principalmente inalámbrica y algunas tecnologías para conectar los dispositivos son: RFID, ADSL, GPRS, Wifi, LAN, 3G. [5]

La gran mayoría de instituciones en todo el mundo integra a los seres humanos desde los dispositivos móviles en una comunidad interactiva proporcionando control y actividad a través del Internet. En los últimos años la aparición de las redes y los medios sociales, como Facebook y Twitter, sumado al crecimiento exponencial del acceso al internet móvil ha transformado el internet de los computadores en el internet de las personas. Cuando las personas navegan en la web, pueden resolver problemas de una forma más eficiente y rápida, desde un ordenador ubicado en cualquier parte del mundo, debido a que el Internet se introdujo como una red global, donde los computadores se conectan con los servicios. [5]

En la actualidad existen aproximadamente 1.5 mil millones de computadoras con acceso a internet y más de 1 millón de teléfonos celulares con conexión a una red móvil, todo este avance tecnológico ha dado lugar al nacimiento de un término nuevo llamado Internet de las Cosas o por sus siglas en Ingles "IoT (Internet of Things)", en el que 50 a 100 mil millones de dispositivos serán conectados a la red para el año 2020. [4]

El rápido desarrollo de las tecnologías del Internet de las cosas hace posible la conexión de varios objetos a través de la interoperabilidad de internet y proporcionan métodos para la recolección de datos para el propósito de

alguna aplicación. Investigaciones recientes muestran las posibles soluciones en la información intensiva industrial en sectores como los servicios de salud, aplicando tecnologías de IoT.

En la última década, se ha realizado un creciente número de investigaciones hacia el uso de IoT, para adquirir datos de forma inalámbrica en el campo de la salud. El Ambient Assisted Living (AAL) es diseñado para apoyar las actividades diarias de las personas de edad avanzada el mayor tiempo posible, un sistema de consultas médicas a través de Internet para los pacientes rurales con especialistas que trabajan en la ciudad. El uso de sensores médicos y tecnologías de la comunicación atrae más a los investigadores quienes actualmente aplican la tecnología móvil de cuarta generación 4G para la atención médica. [4]

Los servicios de salud en el hogar basados en IoT tienen un amplio potencial empresarial. Se ha puesto en funcionamiento una aplicación para el hogar llamada iHome Health-IO, en particular esta plataforma incluye un espacio abierto basado en una caja inteligente de medicina (iMedBox) con conectividad mejorada y capacidad de intercambio para la integración de dispositivos y servicios, envases farmacéuticos inteligentes (iMedPack) con capacidad de comunicación habilitado por radiofrecuencia RFID, y un dispositivo portable con sensor biomédico (Bio-Patch). [6]

En la actualidad la evolución del modelo médico tradicional hacia la medicina participativa puede ser impulsado por el Internet de las Cosas, el cual incluye el uso de sensores en red, ya sea usados en el cuerpo o implantados dentro de los entornos de vida, con el propósito de monitorear el bienestar físico, como también la salud mental del usuario y activar la asistencia remota.

Mediante el empleo de estas tecnologías en las actividades de control de la salud los médicos son capaces de acceder a diferentes tipos de fuentes de datos de manera fácil y rápida, ayudando a tomar decisiones médicas de emergencia, con el fin de reducir costos en el proceso de atención médico-paciente.

Actualmente en el Ecuador, no se conoce de un sistema de monitoreo portable que entregue información de pacientes a través de aplicaciones web, por lo que sería de gran ayuda contar con un sistema que permita controlar y prevenir algún suceso inesperado en su corazón.

El sistema propuesto relaciona salud y tecnología al brindar ayuda a personas que padezcan enfermedades referentes al corazón, el cual consiste en un prototipo portable de fácil uso, que permitirá transmitir datos de la frecuencia cardíaca mediante el uso de telefonía móvil a cualquier parte del mundo a través de Internet. Los datos provienen de la cantidad de luz que se refleja por la sangre en el interior, el detector deja pasar más corriente cuando recibe más luz y a su vez provoca una caída de tensión, representada por la señal del ritmo cardíaco, también se podrá transmitir la ubicación del usuario que se obtiene por medio del sistema GPS. [7] Finalmente se obtendrá un sistema que permita dar seguimiento en tiempo real a los pacientes que padezcan de alguna enfermedad cardíaca mediante una aplicación web.

1.2 Justificación.

En el Ecuador, la hipertensión arterial y la cardiopatía isquémica son las enfermedades cardiovasculares más comunes. De acuerdo con la información proporcionada del Instituto Nacional de Estadísticas y Censos (INEC), se encuentran entre las primeras causas de mortalidad masculina en el 2014, las enfermedades isquémicas con un total de 2.643 fallecimientos, alcanzando el 7,69%. En el mismo año las enfermedades cerebrovasculares ocuparon el segundo lugar como causa de muerte femenina con 1.874 defunciones [8].

Esta propuesta surge debido a la carencia de un dispositivo portable que analice de forma periódica el comportamiento del órgano principal del sistema circulatorio, especialmente para aquellas personas que padezcan algún tipo de enfermedad. El sistema funciona básicamente de manera secuencial, paciente – sensores – dispositivo – aplicación web – médico y/o familiar, con lo cual las facilidades que se encuentran serán de beneficio tanto para el

paciente como para el médico especialista, dando un gran aporte tecnológico, a la telemedicina, telemetría y a la salud.

1.3 Objetivos.

1.3.1 Objetivo general.

Implementar un sistema portable de monitoreo del ritmo cardíaco de una persona mediante sensores, para detectar una anomalía en tiempo real.

1.3.2 Objetivos específicos.

- Determinar el medio más conveniente de lectura del ritmo cardíaco revisando información de diferentes tipos de sensores con el fin de obtener una buena interpretación de la frecuencia cardíaca.
- Implementar un módulo de lectura del ritmo cardíaco mediante el uso de un sensor de luz para obtener la información que se necesita conocer del paciente.
- Desarrollar una aplicación que permita observar los datos del ritmo cardíaco de una persona en tiempo real mediante gráficos, o reportes y que además pueda notificar de alguna anomalía.
- Establecer la comunicación entre las diferentes tecnologías mediante conexión remota protocolo TCP/IP para la transmisión de las señales adquiridas del ciclo cardíaco hacia la plataforma web.
- Crear un diseño del prototipo, de fácil portabilidad y manejo para el paciente, utilizando softwares especializados en diseño 3D, para permitir la colocación de los componentes electrónicos en el interior del dispositivo.

1.4 Alcance y limitaciones.

El proyecto permitirá que cada paciente disponga de un dispositivo lector de ciclo cardíaco y que luego envíe esta información a la web, el cual funcionará desde cualquier lugar donde tenga acceso a Internet, en el momento en que lo necesite (vida cotidiana, internado en un hospital o en una situación de emergencia).

Una de las limitaciones en el prototipo es la parte de comunicación entre el dispositivo del paciente y la red GSM, para lo cual se debe contar con un plan de datos de alguna operadora del país; otro inconveniente que provenga de su uso es el referente a las interferencias existentes, que puedan afectar a la calidad de la señal producida por diferentes factores, por ejemplo, el tipo de construcción de los edificios, naturaleza, blindaje de las ambulancias, entre otras.

Otra limitación será el tiempo de duración de la batería que utilice el dispositivo el cual tendrá que recargarse periódicamente para así evitar que deje de funcionar por falta de energía.

CAPÍTULO 2

2. MARCO TEÓRICO.

En el presente capítulo se describen las diferentes tecnologías aplicadas que son empleadas para la generación de una solución al problema mencionado en el capítulo anterior. Todas estas tecnologías se las ha clasificado en cuatro campos: Componentes Electrónicos, Tecnologías de Comunicación, Tecnologías de Desarrollo y Softwares Complementarios

A continuación, se describen detalladamente cada una de las tecnologías.

2.1 Componentes electrónicos.

La actividad eléctrica del corazón se provoca cuando este late y se generan impulsos eléctricos, existen diversos medios para poder hacer una lectura del ritmo cardíaco, uno de los más conocidos es el electrocardiógrafo, el cual utiliza electrodos fisiológicos que se colocan sobre el cuerpo del paciente para obtener las señales ECG, más conocidas como electrocardiograma, que es una representación gráfica de la actividad eléctrica del corazón. [9]

2.1.1 Sensor de pulso: Pulse Sensor.

Una opción para poder leer la frecuencia cardíaca o los latidos por minuto de una persona, son los sensores de luz, uno de los más conocidos es el *Pulse Sensor*, que se observa en la figura 2.1, el cual se ha escogido debido a su facilidad en la lectura del ritmo cardíaco y a su sencillez en el uso. Consta de un sensor óptico, una etapa amplificadora y una etapa final para filtrar el ruido de la señal, lo cual hace que su señal analógica de salida sea estable y confiable, este módulo pulse sensor, trabaja con una alimentación de entre los 3 a 5 voltios, y un consumo de corriente máximo de 4mA, todas estas características además de su pequeño tamaño, lo hacen conveniente al momento de procesar la señal analógica y convertirla en un valor exacto conocido como BPM (beats per minute) también llamado latidos por minuto. [10]

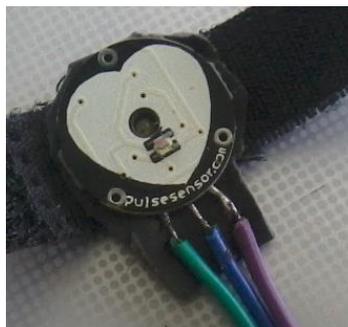


Figura 2.1: Sensor de pulso cardíaco, “Pulse Sensor”, utilizado para realizar la lectura del ritmo cardíaco mediante un led y un fotosensor.

El sensor de pulso que se utiliza en el dispositivo es básicamente un fotopleletismógrafo, que es un instrumento conocido en la medicina para realizar el monitoreo del ritmo cardíaco. La señal que proporciona un fotopleletismógrafo no corresponde a una señal eléctrica como la que se obtiene de un electrocardiograma, sino que es una oscilación en el voltaje analógico. La gráfica de la onda del pulso del corazón se denomina fotopleletismografía o PPG.

Fotopleletismografía.

Antes de explicar lo que es una fotopleletismografía y como se forma su gráfica, se hará una pequeña explicación sobre la representación del ritmo cardíaco en un electrocardiograma (ECG). La cual se puede apreciar en la figura 2.2.

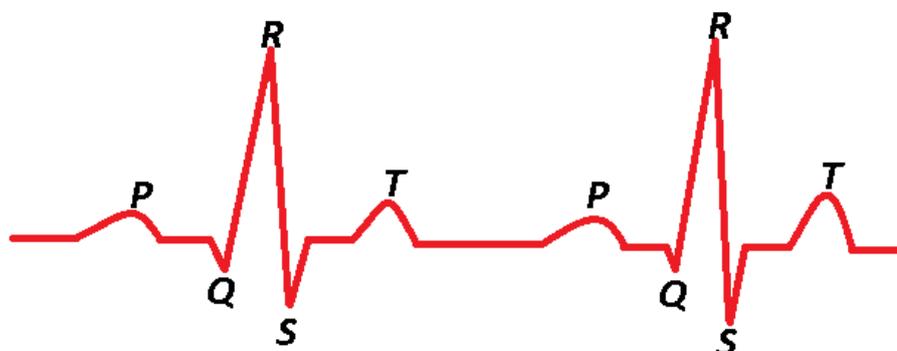


Figura 2.2: Gráfica del electrocardiograma (ECG). [11]

Un electrocardiograma representa el ritmo cardíaco del corazón correspondiente a su señal eléctrica, se la obtiene a través de un electrocardiografo que usando electrodos mide la corriente del corazón. Cuando un electrocardiografo detecta un latido del corazón, se forma un trazado, el cual consiste de una onda P, seguida por un complejo QRS, y una onda T.

A diferencia de un electrocardiograma, la fotopleletismografía proviene a partir de la variación de luz en un tejido capilar, una fuente más que diferente al ECG, sin embargo ambas gráficas tienen un parecido muy lógico, debido a que ambas muestran el momento en el que se produce un latido en el corazón. [12]

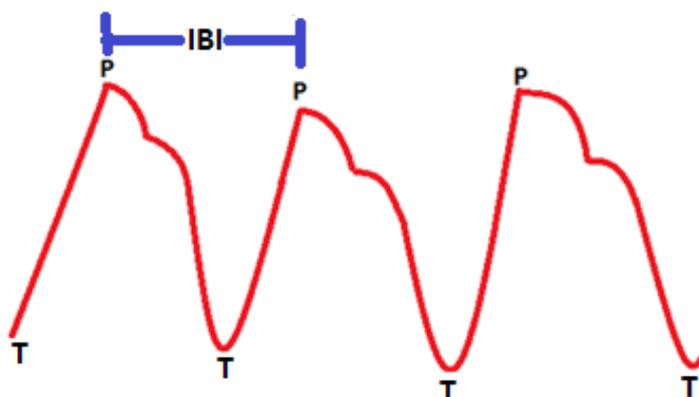


Figura 2.3: Modelo de gráfica de una fotopleletismografía (PPG).

[13]

El trazado de una fotopleletismografía que se ve en la figura 2.3 se modela a partir de tres variables, un valor valle T, que es el valor más bajo del pulso, seguido de un valor pico P, que es el valor más alto del mismo pulso, y por último un valor IBI que representa el intervalo de tiempo entre cada pulso.

2.1.2 Módulo Sim808.

Al momento de realizar el envío de datos tomados por un sensor, uno de los factores más importantes al instante de escoger el sistema de

comunicación, es la distancia a la cual se quiere comunicar, esto quiere decir conocer cuál será el destino donde se quieren enviar los datos, por tal razón, si se desea enviar esta información hacia cualquier punto geográfico dentro de un país, se necesita usar un sistema de comunicación de telefonía móvil, el más utilizado alrededor de todo el mundo incluido Ecuador es la tecnología GSM (*Global System for Mobile communication*), pero si también se plantea poder subir esta información a la web se puede optar por varias tecnologías de internet móvil, la elección de una de estas tecnologías depende mucho del tamaño de los paquetes de datos que se desea transferir a la web, en la actualidad existen 4 generaciones de telefonía móvil habilitadas en el país, mayormente conocidas como 1G, 2G, 3G Y 4G, entre ellas las únicas con capacidad para poder enviar paquetes a la web son la 2G, 3G, y 4G. Dentro de la segunda generación se encuentran las tecnologías GPRS (*General Packet Radio Service*) que tiene una velocidad de carga de 15 a 20kbps en condiciones muy bajas y la tecnología EDGE (*Enhanced Data rates for GSM Evolution*) con una velocidad de carga de 30 a 60 kbps, estas velocidades bastan para poder enviar por internet la información recolectada por un sensor.

Por ventaja de costos, como se detalla en la tabla 1, es más conveniente usar un módulo con características GSM y GPRS, en el mercado existen varios módulos de comunicación que cumplen estas características, el más recomendable para usar es el que produce la empresa SimCom, que ofrece el módulo Sim808, que se contempla en la figura 2.4, este módulo trabaja con una alimentación de 3.4v a 4.4v y una de sus principales ventajas es que funciona en las cuatro bandas 850, 900, 1800 y 1900MHz, que son las frecuencias en las que operan las 3 compañías de telefonía móvil dentro de Ecuador, otra de sus principales características es que viene integrada la conexión para una antena GPS (*Global Position System*). [14]

MÓDULO DE COMUNICACIÓN	TECNOLOGÍA	PRECIO
Sim808	GPRS/GSM	\$ 12,24
Sim908	GPRS/GSM	\$ 26,46
Sim5360A	HSDPA/WCDMA/GSM/GPRS/EDGE	\$ 29,25
Sim7100E	LTE/HSDPA/WCDMA/GSM/GPRS/EDGE	\$ 63,61

Tabla 1: Tabla comparativa de costos de los diferentes módulos de comunicación celular.

Cabe recalcar que los precios que se observan en la Tabla 1, son a nivel internacional, específicamente en China o países de Europa, en los que se incluye el valor de envío a Ecuador.



Figura 2.4: Módulo de comunicación Sim808, elaborado por la compañía SimCom, empleado para realizar el envío de datos del ritmo cardíaco, vía GPRS hacia la aplicación web.

2.1.3 Arduino Pro mini.

Entre los circuitos electrónicos más utilizados alrededor del mundo para realizar el procesamiento de datos adquiridos por un sensor analógico, son los que ofrece la empresa Arduino. Arduino es una plataforma de hardware libre, la cual es básicamente un circuito con un microcontrolador además de un entorno de desarrollo, creada para hacer más sencillo la implementación de diversos proyectos electrónicos que incluyan programación de los mismos.

Entre los diferentes módulos que ofrece Arduino, existe uno muy conveniente para el desarrollo del dispositivo debido a sus pequeñas dimensiones, el Arduino pro mini con una longitud igual a 33.3mm, y un ancho de 18mm, las cuales ayudan a que el tamaño final del prototipo del dispositivo sea lo más pequeño posible.

Existen dos versiones de arduino pro mini, la primera contiene un oscilador de 8 MHz, y la segunda con un oscilador de 16MHz, ambos módulos utilizan un microcontrolador Atmega 328, el cual incluye una memoria Flash de 32 KB, donde se almacena el código del programa, una memoria SRAM de 2 KB, y una memoria EEPROM, de 1 KB.

Se ha elegido la versión de 8 MHz, que incluye un regulador de voltaje S20K por lo que trabaja con un voltaje regulado de funcionamiento de 3.3v, y consume una corriente de tan solo 40mA, puede ser alimentado con un voltaje de 3.35v a 12v, a través de sus pines Vin y Gnd.

El Arduino Pro mini de 8MHz que se observa en la figura 2.5, se puede comunicar fácilmente con un computador, otro arduino, o algún otro microcontrolador diferente, en este caso se comunica muy bien con el módulo de comunicación Sim808, el microcontrolador Atmega328 incluye un puerto serial UART TTL de 3.3 voltios, además se puede aumentar la cantidad de puertos seriales en cualquiera de los pines del Arduino agregando la biblioteca SoftwareSerial en el código del programa.

Su programación es mediante un IDE y un lenguaje de programación propio también llamado arduino, similar a lenguaje de programación C. [15]

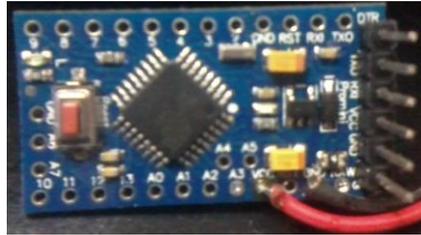


Figura 2.5: Placa de Desarrollo Arduino Pro-mini, utilizado para el procesamiento de la señal tomada por el sensor y luego enviarla hacia el módulo de comunicación.

2.1.4 Batería LiPo.

Al momento de poner en funcionamiento un dispositivo electrónico o un proyecto en general, es indispensable conocer qué tipo de alimentación utiliza y las características de la misma, en este caso se está usando varios elementos electrónicos, entre los cuales existen 2 módulos electrónicos importantes que por sus características sirven de guía cuando se elige el tipo de batería con la que se va a trabajar, el primer módulo importante es el Sim808 que trabaja con un voltaje entre los 3.4v y los 4.4v, y el segundo módulo es el arduino pro mini que necesita una alimentación externa de 3.35v a 12v, debido a que estos dos módulos pueden trabajar con una fuente de voltaje mayor a los 3.4v, la mejor opción es el uso de la batería 103450W, que se ve en la figura 2.6, la cual es una batería de litio y polímero, y entrega un voltaje nominal de 3.7V, y 4.2V cuando se encuentra completamente cargada, con esta batería se puede alimentar en paralelo ambos módulos, esta batería funciona con una carga de 2000mAh, la misma que es suficiente para un tiempo de duración de 12 a 18 horas teóricamente. [16]



Figura 2.6: Batería de Litio Polímero, de 3.7V y 2000mAh, utilizada para energizar cada uno de los elementos electrónicos del dispositivo.

2.2 Tecnologías de comunicación.

La comunicación entre dispositivos se refiere esencialmente al intercambio de información entre ellos, el principal objetivo de las comunicaciones se basa en que cualquier dispositivo pueda transmitir las señales de forma eficaz y segura, hacia otro o más dispositivos. [17]

La comunicación inalámbrica es aquella comunicación sin cables, por lo que la información que se transfiere del emisor al receptor no se realiza por medios de propagación físicos, sino que aprovechan el uso de las ondas electromagnéticas en el espacio, por consiguiente, los únicos dispositivos físicos para una comunicación inalámbrica son las antenas, las cuales se encuentran una en el emisor y otra en el receptor. [18]

2.2.1 Antena Gsm/Gprs.

Para que el módulo Sim808 pueda tener una mayor intensidad en su señal de comunicación GSM o GPRS es necesario el uso de una antena con capacidad de conexión inalámbricamente por ondas de radio, hacia una estación base de telefonía móvil, dependiendo de la banda en la que se encuentre trabajando el módulo Sim808, una buena opción es el uso de las antenas de material flexible o de tipo sticker o pegatina, que se puede apreciar en la figura 2.7, ya que se

colocan fácilmente dentro de cualquier dispositivo, también se debe tener en cuenta que estas antenas tienen una impedancia de 50 ohmios el cual es uno de los principales requisitos que exige el módulo Sim808 para ser conectadas, otra característica importante para ser elegidas se basa el valor de su ganancia que es igual a 2db trabajando en cualquiera de las cuatro bandas, estas antenas trabajan muy bien en tecnologías GSM, GPRS, que son las que se usan para enviar información hacia algún teléfono móvil o hacia un servidor web. [19]



Figura 2.7: Antena GSM/GPRS con conector ufl, empleada para aumentar la intensidad en la señal de comunicación GPRS.

2.2.2 Antena Gps.

Como se menciona en el capítulo anterior, una característica adicional que brinda el módulo Sim808 es el uso del GPS, el cual es de mucha utilidad para obtener las coordenadas de la ubicación de un paciente, se precisa monitorear el ritmo cardíaco y el lugar donde se encuentra, esto es un beneficio adicional que ayude a resolver el problema planteado en el capítulo 1, para esto es necesario conocer las consideraciones al momento de elegir una buena antena GPS, la primera característica que cumple la antena que se observa en la figura 2.8, es tener una impedancia de 50 ohmios que es lo permitido por el módulo Sim808, otra característica que se ha tomado en cuenta es su pequeño tamaño al tener dimensiones menores a los 9mm x

9mm, pero su característica de mayor importancia es trabajar con una ganancia de 2db, la cual es mayor al valor mínimo recomendado por el módulo de comunicación Sim808 de -4db. [20]

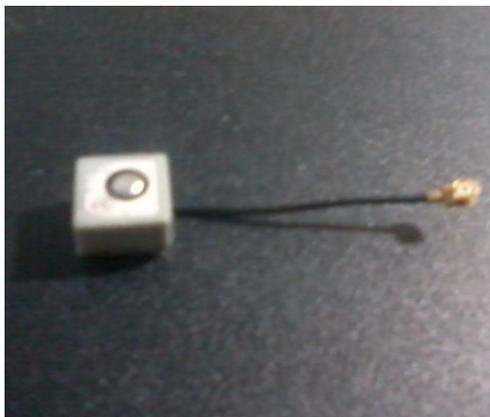


Figura 2.8: Antena GPS Pasiva con conector ufl, utilizada para obtener las coordenadas de la posición geográfica del dispositivo.

2.3 Tecnologías de desarrollo.

Cuando se plantea la creación de un sistema completo de telemedicina, donde se transfieren datos correspondientes al ritmo cardíaco desde un dispositivo hasta una aplicación web, se debe trabajar con tecnologías de desarrollo, tanto para la programación del dispositivo como para la creación de la aplicación web.

El desarrollo de aplicaciones web ha evolucionado notablemente en los últimos diez años, desde el punto de vista del desarrollo de software como también a nivel de la administración de sistemas. Se han creado decenas de tecnologías, frameworks de desarrollo de aplicaciones, librerías, herramientas, arquitecturas para desarrollar una aplicación web. [21]

El internet y las aplicaciones web han logrado mejorar en gran manera la administración de sistemas en muchos aspectos. Para que una aplicación

web trabaje correctamente necesita contar con una base de datos y un servidor web.

2.3.1 IDE de Arduino 1.6.11.

El Entorno de Desarrollo Integrado o IDE es un software de código abierto que hace más fácil escribir el código y luego subirlo a la placa.

Es un software multiplataforma, se ejecuta normalmente en cualquier sistema operativo como Windows, Mac OS y cualquier distribución de Linux. Esta desarrollado en Java y basado en Processing y otro software de código abierto. Este entorno de desarrollo es compatible con cualquier módulo Arduino. [22]

El lenguaje de programación que utiliza Arduino se basa mayormente en C, y permite utilizar todas las funciones del estándar C y algunas del C++. En la figura 2.9 se puede visualizar una pantalla del IDE de Arduino.

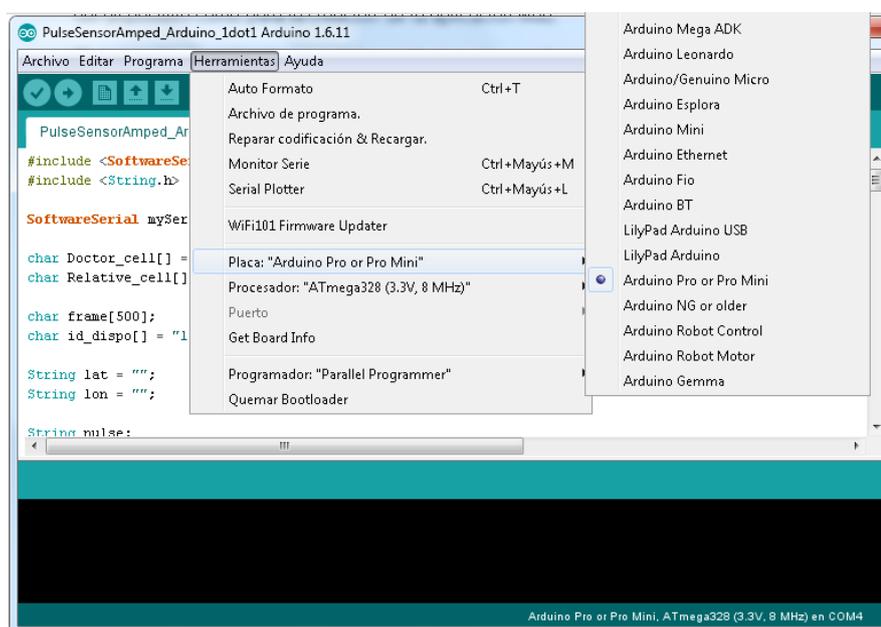


Figura 2.9: Plataforma de desarrollo arduino.

2.3.2 Django.

Django es un framework web de alto nivel de Python que fomenta el rápido desarrollo y diseño limpio y pragmático. Creado por desarrolladores con experiencia, que se encarga de gran parte del trabajo de desarrollo Web, para que el desarrollador pueda centrarse en la escritura de su aplicación sin necesidad de reinventar lo que ya existe. Una ventaja de Django es que es de código abierto y libre.

El solo hecho de ser un framework ya es una ventaja a favor de Django, el cual fue diseñado para ayudar a que los desarrolladores puedan crear sus aplicaciones lo más rápidamente posible en cuestión de días e incluso en horas, con Django se puede tener una aplicación bastante funcional y a medida que se va aprendiendo, se puede llegar a reducir aún más el tiempo de desarrollo.

Vale la pena recalcar también que además de las características antes mencionadas, Django también cuenta con sistemas de seguridad, adicional trabaja con un servidor web incluido para realizar pruebas, la meta fundamental de Django es facilitar la creación de sitios web completos ya que cuenta con una amplia documentación, que facilita el rápido desarrollo con módulos pocos dependientes y amigable en diseño, con un principio fundamental DRY que significa “no te repitas!” utiliza esta filosofía para no crear bloques de códigos iguales y fomentar la re-utilización de código. [23]

2.3.3 Python.

Al ser Django un framework que utiliza como lenguaje de desarrollo Python, hace necesario su instalación y uso en el desarrollo de aplicaciones por medio de Django. Python es un lenguaje de programación de alto nivel, interpretado, es un lenguaje dinámico ampliamente utilizado. Su filosofía de diseño hace hincapié en la

legibilidad del código, y su sintaxis permite a los programadores expresar conceptos en menos líneas de código que otros lenguajes.

Python soporta múltiples paradigmas de programación, incluyendo la programación orientada a objetos, programación imperativa y programación funcional. Cuenta con un sistema de tipo dinámico y gestión automática de memoria y tiene una librería estándar grande y amplia. [24] En la figura 2.10 se aprecia una ventana correspondiente a Python.

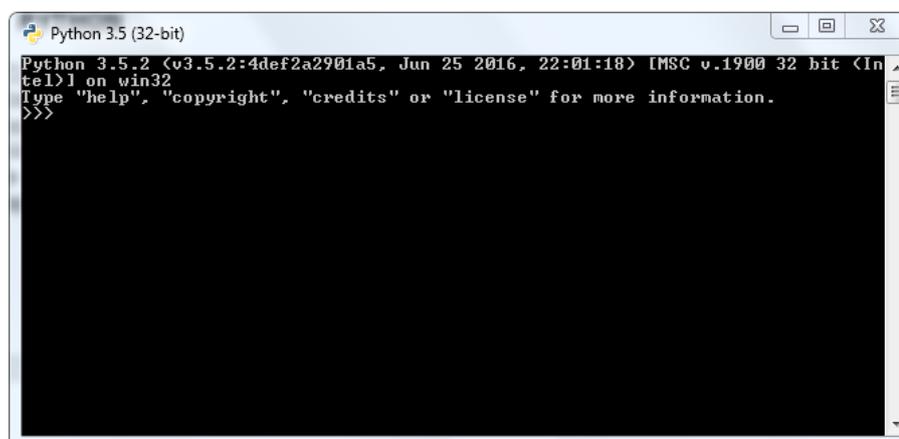


Figura 2.10: Consola interactiva de Python.

2.3.4 Html.

HTML es un lenguaje de marcas de hipertexto para editar documentos web, más conocidas como páginas web, este lenguaje es un conjunto de etiquetas de marcado llamadas etiquetas *HTML*, cada una de las etiquetas sirven para describir los diferentes contenidos de los documentos.

Es un estándar que define el contenido de una página web como el texto, las imágenes, los videos, entre otros mediante una estructura básica y un código HTML. Es considerado el lenguaje web más utilizado a nivel mundial el cual ha sido impuesto para visualizar las páginas. [25]

2.3.5 Apache.

Apache es un servidor HTTP de código abierto para sistemas operativos modernos, incluyendo UNIX y Windows, más utilizado en el mundo. El objetivo de Apache es proporcionar un servidor seguro, eficiente y extensible que proporciona servicios HTTP en sincronización con los estándares HTTP actuales.

Entre las ventajas del Servidor Apache sobre otros servidores web, las que se puede nombrar es su alto nivel de configuración, su licencia gratuita y su compatibilidad con los sistemas operativos al ser multiplataforma. [26]

2.3.6 MySQL.

Al momento de desarrollar una aplicación web que maneje datos de diferentes usuarios o en este caso pacientes, se debe dedicar parte del desarrollo a la administración del sistema, es necesario contar con un sistema de gestión de base de datos. MySQL es la opción de base de datos más popular en el mundo para ser usada en aplicaciones web, es un sistema para la administración de base de datos relacional de código abierto. El cual está disponible bajo los términos de la Licencia Pública General de GNU.

Una de las mayores ventajas de MySQL es su velocidad al realizar las operaciones, lo que lo hace uno de los gestores de base de datos con mejor rendimiento, una segunda ventaja es su facilidad al momento de su instalación y configuración ya que soporta gran cantidad de sistemas operativos, otra ventaja importante es que debido a su bajo consumo puede ser ejecutado en una maquina con escasos recursos sin ningún problema. [27]

2.4 Software complementario.

En esta sección se describen aquellos software que son empleados para complementar el desarrollo tanto del sistema como del dispositivo.

2.4.1 Autodesk Inventor Professional.

Autodesk Inventor es un software desarrollado por la compañía Autodesk, creador también de AutoCAD, es una aplicación de diseño asistido por computador para la creación de prototipos digitales en 3D, visualización y simulación de productos.

Autodesk Inventor se basa en técnicas de modelado paramétrico, no debe ser confundido con los programas tradicionales de CAD. Inventor es utilizado en el diseño de ingeniería, para la creación y producción de nuevos productos. Permite modelar las dimensiones geométricas, esto permite que el creador pueda aplicar sus conocimientos de cálculo dentro del diseño.

Una de las ventajas principales de Autodesk Inventor, respecto a otros softwares de diseño asistido por computadora como SolidWorks, es que permite trabajar con una licencia gratuita de tipo educativa autónoma, durante 3 años, hasta en un máximo de 2 dispositivos, lo cual evita costos económicos en adquisición de software. [28] Un ejemplo del área de Diseño en Autodesk Inventor se observa en la figura 2.11.

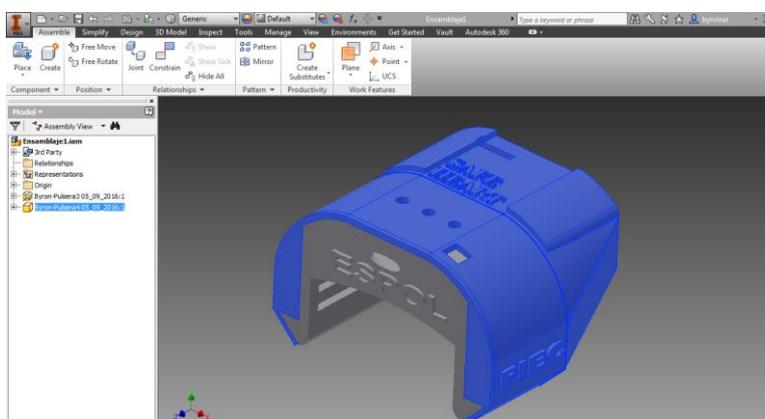


Figura 2.11: Área de diseño en Autodesk Inventor Professional.

2.4.2 Lucidchart.

LucidChart es una herramienta web, que ayuda a crear diagramas a través de una interfaz de navegador con soporte HTML5, característica común de todos los navegadores modernos, no necesita instalación de ningún software. Permite construir diagramas de flujo, mapas mentales y organizadores gráficos. Su uso es muy sencillo, solo se necesita arrastrar y soltar cada elemento de un gráfico. Tiene la capacidad de permitir a varios usuarios trabajar al mismo tiempo, ya que se actualiza constantemente.

Esta herramienta se puede utilizar de forma gratuita, previa creación de una cuenta, lo cual ayuda a guardar y publicar los diagramas que se van realizando. [29] En la figura 2.12 se puede observar una ventana del área de trabajo en Lucidchart.

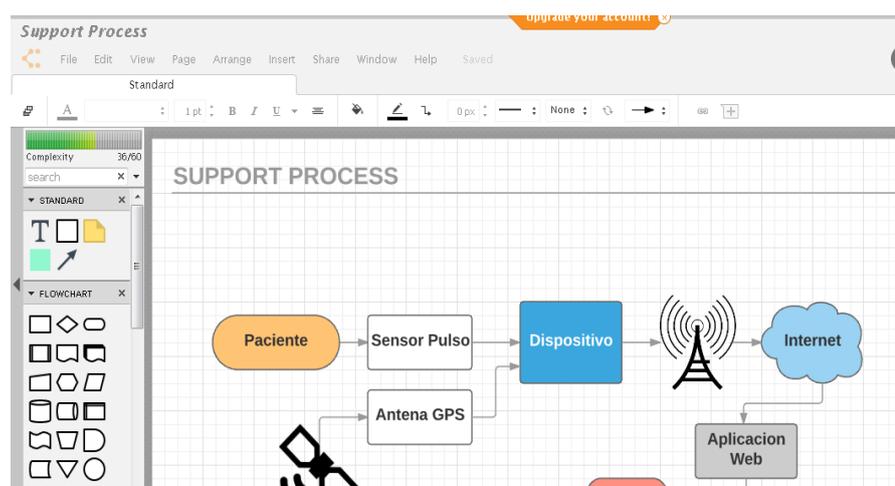


Figura 2.12: Área de trabajo de Lucidchart, empleado para los diagramas de flujo.

CAPÍTULO 3

3. IMPLEMENTACIÓN Y DESARROLLO.

En el capítulo actual se presenta el sistema propuesto para solucionar el problema planteado en el primer capítulo. También, en este capítulo se detalla cada bloque que conforma el sistema implementado como solución. El sistema funciona básicamente de manera secuencial, paciente – dispositivo – servidor – aplicación web – médico, como se muestra en la figura 3.1.

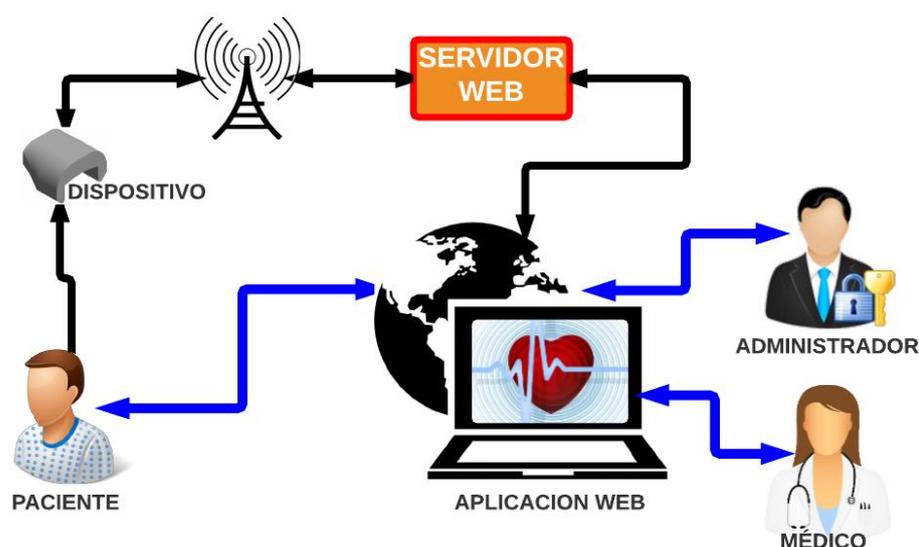


Figura 3.1: Diagrama de bloques general del sistema.

A continuación, se describen detalladamente cada una de los bloques que integran el sistema.

3.1 Paciente.

Este bloque se refiere al usuario del dispositivo, es la persona a la cual se tomará la información respectiva a la actividad de su corazón, se estima que los pacientes sean personas con alguna anomalía cardíaca, o personas de edad avanzada que requieran monitoreo continuo del funcionamiento de su sistema cardíaco, aunque el dispositivo puede ser usado por cualquier persona, sean estos niños o adultos que estén dispuestos a hacerlo.

3.2 Dispositivo.

El dispositivo es uno de los bloques más importantes de todo el sistema, debido a que se trata esencialmente del hardware, el cual es usado por el paciente. Este bloque está conformado de dos partes necesarias, como lo son el diseño estético de la carcasa en su exterior y su parte electrónica interna.

3.2.1 Diseño de la carcasa.

Para determinar el diseño más útil del dispositivo, se debe partir de varios aspectos, entre ellos, su forma de uso, su portabilidad, y el lugar donde será colocado en el paciente, se plantea que el dispositivo tenga un diseño similar a una pulsera para poder medir su pulso desde la muñeca del paciente o su dedo índice, debido a q estos 2 puntos son lugares de alta efectividad al momento de realizar la lectura del ritmo cardíaco a través de un sensor infrarrojo.

Por lo tanto, el diseño está conformado por dos partes, las cuales son su base y su cobertor. En la figura 3.2 se puede apreciar el diseño del prototipo ensamblado. En el Anexo B, se describen las medidas de las dos partes del diseño con sus respectivas vistas.

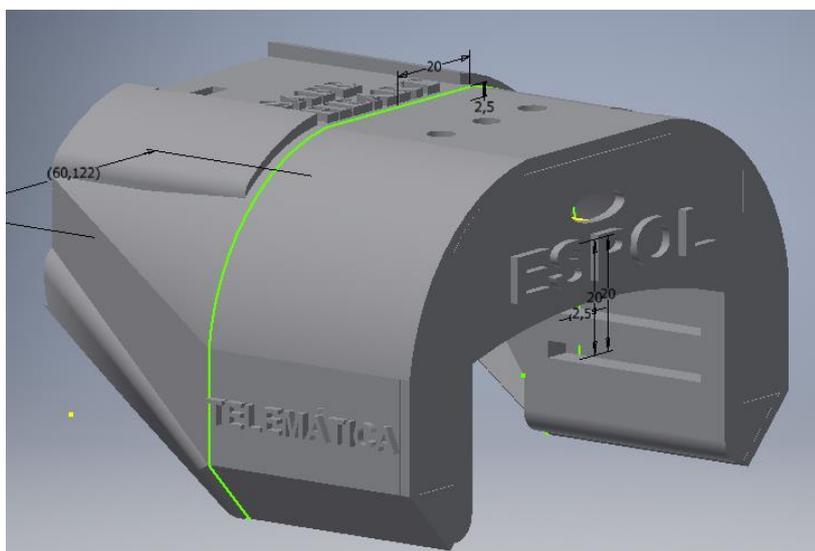


Figura 3.2: Diseño de la carcasa en vista lateral.

Diseño de la base de la carcasa.

El diseño de la base que se muestra en la figura 3.3, es definido debido a 2 características que esta cumple. La primera característica es que la base es la parte del dispositivo que se coloca en la parte superior de la muñeca del brazo del paciente, por ello deberá tener una abertura estándar para poder encajar en la muñeca de cualquier persona, las medidas de la abertura definidas son de 70mm de ancho y 26mm de altura, en esta abertura se ubica la correa, que cumple la función de sujetar el dispositivo a la muñeca del brazo del paciente. La segunda característica que debe cumplir es la que su nombre indica, ya que es la base donde se colocan todos los elementos electrónicos del dispositivo, para lo cual se establece un espacio físico de 80mm de longitud, 60mm de ancho y 20mm de altura.

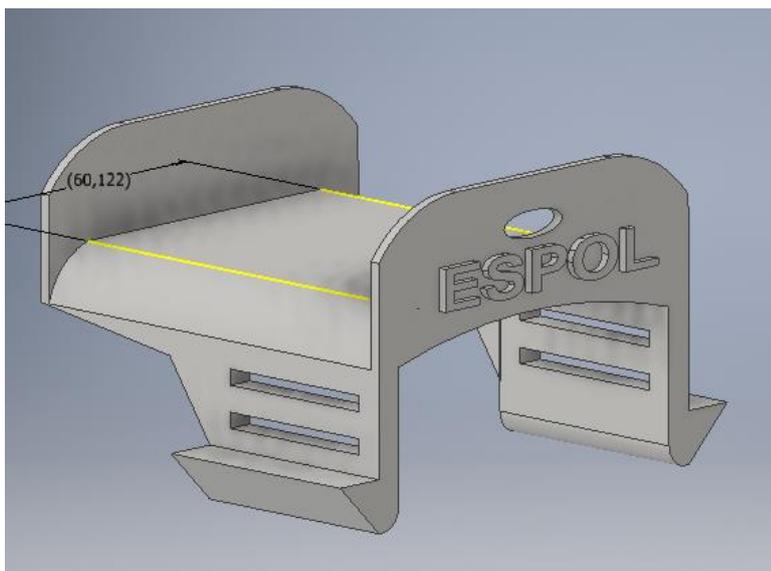


Figura 3.3: Diseño de la base de la carcasa.

Diseño del cobertor de la carcasa.

El diseño del cobertor que se ve en la figura 3.4, se basa principalmente en su función de proteger los elementos electrónicos

dentro del dispositivo, y así mismo en la parte estética de presentación del dispositivo.

Esta parte de la carcasa en su lado superior contiene tres orificios circulares de 4.8 mm de diámetro cada uno, en los cuales se muestran leds de diferentes colores, para indicar tres señales distintas enviadas desde el módulo principal, además contiene un orificio cuadrado de 9mm por lado, en este orificio está ubicado el interruptor de encendido y apagado del dispositivo, y por ultimo incluye un orificio rectangular de 16mm de largo por 3mm de ancho, por el cual se ingresa el conector del cable de programación y configuración del dispositivo.

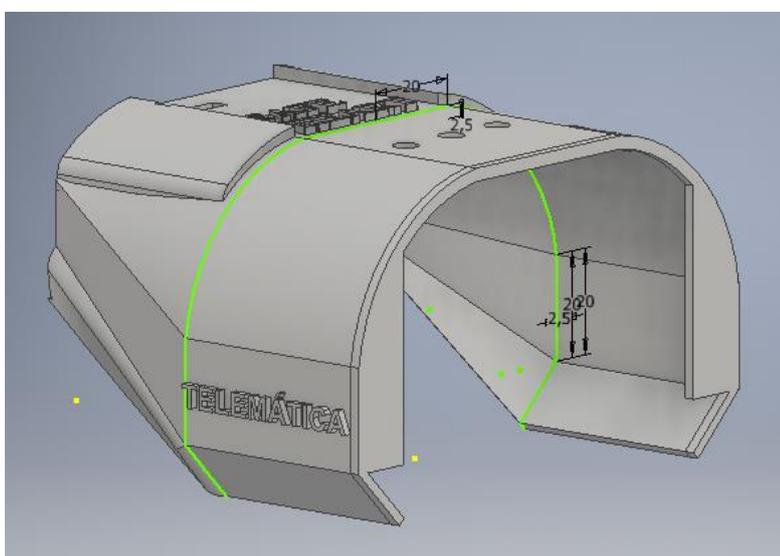


Figura 3.4: Diseño del cobertor de la carcasa.

3.2.2 Electrónica del dispositivo.

La parte electrónica del dispositivo está conformado de 6 elementos importantes, el sensor de pulso, la placa arduino pro mini, el módulo Sim808, la antena GPRS, la antena GPS y la batería de 3.7V, los cuales conectadas entre sí realizan una función importante dentro del sistema, el arduino pro mini que funciona como controlador del dispositivo, el cual recibe los datos del sensor de pulso y los procesa

para luego enviarlos a la web por medio del módulo Sim808 a través de su antena GPRS, que entrega de forma inalámbrica los datos del ritmo cardíaco a la estación base más cercana, y así poder llegar al servidor web.

Otros elementos electrónicos con los que consta el dispositivo son los tres leds de diferentes colores, que indican tres señales diferentes, el primer led es de color verde, el cual indica cada latido del corazón, el segundo led de color rojo que se enciende siempre y cuando exista una alerta respecto al ritmo cardíaco, y el tercer led de color amarillo el cual se activa cada vez que el dispositivo este encendido. Un último elemento electrónico es el interruptor de encendido y apagado de todo el dispositivo.

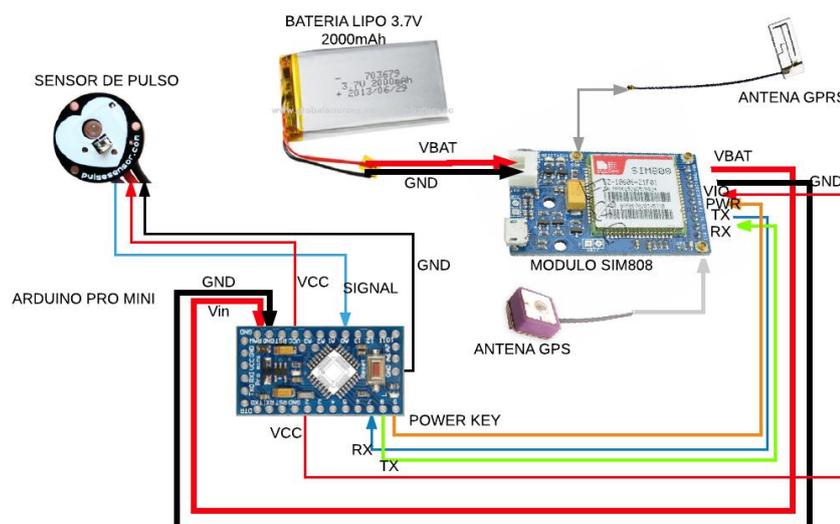


Figura 3.5: Diagrama esquemático electrónico del dispositivo.

Uso del módulo Arduino Pro mini.

El elemento principal del diagrama esquemático electrónico del dispositivo que se muestra en la Figura 3.5, es el módulo Arduino pro mini de 8MHz, que es el controlador de todo el sistema electrónico.

El arduino pro mini de 8 MHZ, contiene 6 entradas analógicas, se utiliza una de estas entradas para recibir la señal del sensor de pulso, así también el sensor se conecta a las señales Vcc y Gnd del arduino

El módulo arduino además puede trabajar con 14 entradas y salidas de señales digitales, de las cuales para este dispositivo se han usado 7 pines, 3 de ellos conectados a los leds indicadores, una salida digital que sirve de señal Vcc que se conecta al pin VIO del módulo de comunicación sim808, 2 pines los cuales se configuran como RX y TX para realizar la comunicación serial, los mismos que se conectan de forma cruzada con los pines RX y TX del módulo sim808, y un último pin digital configurado como señal de salida llamado Power Key, que se conecta al pin PWR del módulo de comunicación.

Todos los pines de la placa arduino que se usan para las conexiones a los diferentes elementos, deben ser definidos al comienzo del programa y configurados dentro de la función setup (), como se muestra en la figura 3.6.

```

SoftwareSerial mySerial (7, 8);
int pulsePin = A0;
int blinkPin = 13;
int VCC = 2;
int Powerkey = 9;

void setup () {
  pinMode (Powerkey, OUTPUT);
  power ();
  pinMode (blinkPin, OUTPUT);
  pinMode (VCC, OUTPUT);
  digitalWrite (VCC, HIGH);
  mySerial.begin(2400);
  Serial.begin(19200);
  delay (500);
  configuracion_inicial ();
  Serial.println("COMUNICACION GPRS HABILITADA");
  interruptSetup ();
}

```

Figura 3.6: Asignación y configuración de pines digitales y analógicos usados en el módulo Arduino.

Uso del sensor de pulso

El sensor de pulso es el elemento electrónico que tiene contacto directo con el paciente, por medio del cual se toma la información de su actividad cardíaca. En la parte frontal del sensor donde se encuentra un dibujo en forma de corazón, existe un led y un sensor de luz ambiente, esta parte frontal es la que se coloca en alguna parte translúcida del cuerpo como la yema del dedo índice, o el lóbulo de la oreja, detecta el paso de la sangre cada vez que el corazón late, por lo que la información que se obtiene con este sensor es la cantidad de latidos por minuto o por sus siglas en inglés el valor BPM, este sensor está sujeto en el dedo con una cinta de velcro. [30]

En el momento en el que el corazón bombea sangre, genera una onda de pulso, la cual se traslada a través de las venas del tejido capilar, que es donde se coloca el sensor de pulso; cuando la onda pasa, genera cambios de intensidad de luz debido al bloqueo de la sangre a la luz emitida por el led que es percibida por el fotosensor. [31] El valor de la señal analógica enviada por el sensor hasta el arduino se encuentra entre 0 y 1024 bits.

Este sensor tiene 3 cables diferentes, el primero transmite la señal analógica que recoge el sensor del pulso cardíaco que se conecta a la entrada analógica A0 del arduino pro mini, el segundo cable corresponde al valor Vcc del cual se alimenta, este valor Vcc puede ir desde los 3v hasta los 5v, por lo que va conectado al pin Vcc del arduino que equivale a un voltaje de 3.3 v; y el tercer cable se conecta al pin Gnd o tierra del circuito.

El sensor reconoce las ondas sucesivas que corresponden a cada latido, y así mismo, calcula el intervalo de tiempo IBI (Inter-Beat Intervale), que existe entre cada uno, utilizando el patrón de onda del PPG.

El funcionamiento del sensor de pulso que se conecta al Arduino, se basa en el código que se escribe en el programa que va dentro del

microcontrolador del Arduino. Se configura una interrupción cada 2ms, que origina una frecuencia de muestreo de 500 Hz. El código llama a la función ISR (), que busca cada 2ms si en la entrada analógica A0 ha habido un pulso. La variable donde se almacena el valor de la señal analógica se denomina Signal. Luego se guardarán los valores más altos de la señal en la variable P que se los ha llamado picos, y los valores más bajos en la variable T que se denominan valles, la diferencia entre P y T se la llama amplitud de la señal. Se inicializa la variable thresh con un valor de 512, thresh es el umbral de la señal, que es la mitad del valor de la amplitud más el valor de T, como se observa en la figura 3.7. [30]

```

void interruptSetup () {
  TCCR2A = 0x02;
  TCCR2B = 0x05;
  OCR2A = 0x7C;
  TIMSK2 = 0x02;
  sei ();
}

ISR(TIMER2_COMPA_vect) {
  Signal = analogRead(pulsePin);
  sampleCounter += 2;
  int N = sampleCounter - lastBeatTime;

  //búsqueda del pico y del valle
  if (Signal < thresh && N > (IBI/5) *3) {
    if (Signal < T) {
      T = Signal;
      VALLE=T;
    }
  }

  if (Signal > thresh && Signal > P) {
    P = Signal;
    PICO=P;
  }
}

```

Figura 3.7: Lectura y almacenamiento de las variables PICO y VALLE que forman la fotopleletismografía.

Para buscar la presencia de un latido se tendrá la condición de no realizar esta búsqueda mientras N sea menor a 250ms, debido a que esto significaría que el corazón estaría latiendo demasiado rápido, y sería demasiado raro encontrar una frecuencia cardíaca mayor a 240 latidos por minuto. Luego de superar esta condición se debe verificar si la señal supera el valor del umbral, y N es mayor a 3/5 del ultimo IBI, significa que ha existido un latido del corazón y se encenderá el led del pin 13 del Arduino, como se aprecia en la figura 3.8. [30]

```

if (N > 250) {
  if ((Signal > thresh) && (Pulse == false) && (N > ((IBI/5) *3)) {
    Pulse = true;
    digitalWrite (pulsePin, HIGH);
    IBI = sampleCounter - lastBeatTime;
    lastBeatTime = sampleCounter;
  }
}

```

Figura 3.8: Código de encendido del led de pulso cada vez que el sensor detecta un latido.

El siguiente código se lo tiene para calcular el tiempo que transcurre entre un primer latido y un segundo latido, el cual corresponde al valor de la variable IBI, que se va almacenando por 10 ocasiones dentro de un arreglo llamado rate, como se ve en la figura 3.9. [30]

```

if(secondBeat) {
  secondBeat = false;
  for (int i=0; i<=9; i++) {
    rate[i] = IBI;
  }
}
if(firstBeat) {
  firstBeat = false; |
  secondBeat = true;
  sei ();
  return;
}

```

Figura 3.9: Código que calcula el tiempo que existe entre dos latidos y se almacena en la variable IBI.

Para realizar el cálculo de los BPM, se inicializa una variable `runningTotal` con un valor igual a 0, para luego guardar dentro de esta el valor del promedio de los últimos 10 IBIs. El valor de los BPM se lo determina dividiendo la cantidad de milisegundos equivalentes a un minuto para el valor final de `runningTotal`. [30] Luego se almacena el valor de `runningTotal` en una nueva variable llamada `TIEMPO`, esto se puede observar en el código que se detalla en la figura 3.10.

```

word runningTotal = 0;
for (int i=0; i<=8; i++) {
    rate[i] = rate[i+1];
    runningTotal += rate[i];
}
rate [9] = IBI;
runningTotal += rate [9];
runningTotal /= 10;
BPM = 60000/runningTotal;
TIEMPO=runningTotal;
QS = true;
}

```

Figura 3.10: Código que calcula el valor de la cantidad de latidos por minuto y la almacena en la variable BPM.

Cuando se detecta que la señal es menor que el valor del umbral se procede a apagar el led del pin 13. Luego se calcula y se asigna los valores a cada una de las variables como la amplitud y el umbral, también se calcula el valor de una variable nueva denominada `NORMAL` que es un valor estable en el ritmo cardíaco antes de empezar un nuevo pulso, como se muestra en la figura 3.11. [30]

```

if (Signal < thresh && Pulse == true) {
    digitalWrite (13, LOW);
    Pulse = false;
    amp = P - T;
    thresh = amp/2 + T;
    NORMAL= amp/4 + T;
    P = thresh;
    T = thresh;
}

```

Figura 3.11: Código que calcula el valor de la amplitud y el umbral de la señal PPG.

Si durante 2,5 segundos no se detecta ningún latido, entonces el programa reinicia todas las variables. [30] esta configuración se puede ver en el código que se describe en la figura 3.12.

```

if (N > 2500) {
    thresh = 512;
    P = 512;
    T = 512;
    PICO=P;
    VALLE=T;
    NORMAL=thresh;
    TIEMPO=2500;
    firstBeat = true;
    secondBeat = false;
    lastBeatTime = sampleCounter;
}

```

Figura 3.12: Código que reinicia las variables cada vez que no se detecta algún latido en un tiempo mayor a 2,5 segundos.

Luego de obtener los valores de todas las variables de importancia como el BPM, el valor del PICO, el valor del VALLE, su valor NORMAL y el valor del TIEMPO, se los envía al servidor WEB, a través la función SubmitHttpRequest (), como se muestra en el código de la figura 3.13.

```
void sensar () {  
    if (QS == true) {  
        SubmitHttpRequest (BPM, PICO, VALLE, NORMAL, TIEMPO);  
        QS = false;  
    }  
    delay (20);  
}
```

Figura 3.13: Envío de datos hacia la función SubmitHttpRequest.

Uso del módulo Sim808.

El módulo sim808 es uno de los circuitos más utilizados para agregarle la funcionalidad de comunicación GSM y GPRS a dispositivos que usen arduino o microcontroladores.

En el módulo viene integrado un conector UFL para antenas GPS, ya sea pasiva o activa, además contiene otro conector UFL aparte para la antena GSM, para recibir la señal de la operadora celular.

La alimentación del módulo de comunicación, se recibe a través de un conector XH-2.0mm, el voltaje de entrada es de 3.4V a 4.4V, lo que hace que sea conveniente conectarlo a una batería Li-Po de 3.7V.

Contiene también un Soporte de tarjeta SIM para una tarjeta de tamaño estándar.

La comunicación entre el arduino y el módulo sim808 se la realiza mediante comando AT, los cuales se pueden encontrar en el manual de usuario del Sim808.

Lo primero que se debe hacer para poner en funcionamiento el módulo Sim808 es encenderlo. Esto se lo hace por medio de la función power(void), que a través del pin Powerkey, el cual permanece energizado por tres segundos, y luego se desactiva, para hacer una espera de diez segundos que permite al módulo captar la señal

correspondiente a la operadora telefónica que se esté usando. El código de esta función se puede observar en la figura 3.14.

```
void power(void)
{
    digitalWrite (Powerkey, HIGH);
    delay (3000);
    digitalWrite (Powerkey, LOW);
    delay (10000);
}
```

Figura 3.14: Función power, que enciende el módulo Sim808.

Luego de que el módulo Sim808 se enciende, debe ser configurado inmediatamente para poder aprovechar al máximo sus funcionalidades como el envío de datos a la web usando la antena GSM/GPRS, y el uso de la antena GPS para determinar la posición geográfica del paciente. La configuración del módulo sim808 se la realiza mediante comando AT, los cuales se puede encontrar en el manual de usuario del Sim808.

El comando "AT+IPR=2400", establece el valor en baudios para la comunicación serial entre el Arduino y el módulo Sim808, el siguiente comando "AT+CSQ" muestra la calidad de la señal detectada por la antena. El comando "AT+CMGF=1" habilita el envío de texto hacia el módulo, para determinar que el tipo de conexión de la antena es GPRS, se envía el comando "AT+SAPBR=3,1, \"CONTYPE\", \"GPRS\"", luego se debe configurar la dirección del servidor APN que corresponde a la operadora de la tarjeta SIM, para poder usar la conexión GPRS enviando el comando "AT+SAPBR=3,1, \"APN\", \"internet.movistar.com.ec\"", finalmente se debe inicializar la comunicación HTTP a través del comando "AT+HTTPINIT". De la misma manera se envían comandos para habilitar el uso del GPS en la tarjeta, los comandos son los siguientes, el comando "AT+CGNSPWR=1" y el comando "AT+CGNSSEQ=RMC" que luego

de ser enviado existe una espera de 45 segundos para que la conexión GPS se establezca. Todos los comandos detallados anteriormente se los envía a través de una función llamada `configuracion_inicial ()` que se muestra en la figura 3.15.

```

void configuracion_inicial ()
{
  mySerial.println("AT+IPR=2400");
  delay (300);
  ShowSerialData ();

  mySerial.println("AT+CSQ");
  delay (100);
  ShowSerialData ();

  mySerial.println("AT+CMGF=1");
  delay (2000);
  ShowSerialData ();

  mySerial.println("AT+CGATT?");
  delay (100);
  ShowSerialData ();

  mySerial.println("AT+SAPBR=3,1, \"CONTYPE\", \"GPRS\"");
  delay (1000);
  ShowSerialData ();

  mySerial.println("AT+SAPBR=3,1, \"APN\", \"internet.movistar.com.ec\"");
  delay (4000);
  ShowSerialData ();

  mySerial.println("AT+SAPBR=1,1");
  delay (2000);
  ShowSerialData ();

  mySerial.println("AT+HTTPIPINIT");
  delay (2000);
  ShowSerialData ();

  mySerial.println("AT+CGNSPWR=1");
  delay (2000);
  ShowSerialData ();

  mySerial.println("AT+CGNSSEQ=RMC");
  delay (45000);
  ShowSerialData ();
}

```

Figura 3.15: Configuración del módulo Sim808.

Uso de la antena Gps.

El bloque llamado Antena GPS se basa en un solo elemento el cual recibe mediante triangulación de satélites las coordenadas geográficas donde se encuentra situado un objeto o persona, esta información es enviada hacia el dispositivo cada vez que este la requiera.

En este caso la antena GPS se encuentra conectada al módulo Sim808 por medio de un conector UFL, por lo tanto, la consulta de la posición geográfica en tiempo real se la realiza constantemente llamando a la función GPS (), la cual retorna una cadena con toda la información necesaria respecto a la ubicación, como se aprecia en la figura 3.16.

```
String GPS () {
  while (1) {
    mySerial.println("AT+CGNSINF");
    ShowSerialData ();
    String response = "";
    long int time = millis ();
    while ((time + 1000) > millis ())
    {
      while (mySerial.available())
      {
        char c = mySerial.read();
        response += c;
      }
    }
    return (response);
    delay (10000);
  }
}
```

Figura 3.16: Función GPS, que realiza la búsqueda de las coordenadas por parte de la antena.

Uso de la antena Gsm / Gprs.

En este caso se usa una antena GSM con conector UFL hembra conectado al módulo Sim808. La antena GSM tiene una ganancia de 2db trabajando a cualquiera de las cuatro bandas dentro del país, lo cual permite enviar mensajes de texto a través de la red GSM y además realizar el envío de los datos hasta el servidor web usando la tecnología GPRS. La transferencia de los datos al servidor web se lo realiza enviando los comandos hacia el módulo Sim808, como se detalla en la figura 3.17.

```
mySerial.print("AT+HTTTPARA=\"URL\", \"http://heartsafe.pythonanywhere.com/mapa/conexion/?");
printf (frame, "id_dispo=%s&pulso=%s&latitud=%s&longitud=%s", id_dispo,
        |pulso, latitud, longitud, picos, valles, estable, periodo);
mySerial.print(frame);
mySerial.println("");
delay (1000);
```

Figura 3.17: Código que realiza el envío de todos los valores hacia el servidor web.

Uso de la batería LiPo.

El uso de la batería permite que el dispositivo pueda ser portable y móvil, independiente de cualquier tipo de fuente inmóvil de voltaje, esta batería proporciona una tensión de 3.7v a 2000mAh la cual alimenta de forma paralela tanto a la placa arduino como al módulo de comunicación Sim808, y por consiguiente a los demás elementos que reciben alimentación directamente de estas dos placas principales.

3.3 Servidor web.

Se utiliza un hosting gratuito dedicado únicamente a aplicaciones basadas en Python, se llama "PythonAnyWhere", es un servidor de entorno amigable, con

acceso bash mediante la web, que incluye un entorno virtual gráfico, ha sido diseñado especialmente para Django y otros Framework basados en Python, que es de ayuda a la hora de despegar la aplicación al internet, y brinda hasta tres meses gratuitos para realizar pruebas. [32] En la figura 3.18 se puede observar la pantalla de configuración gráfica del servidor web.

The screenshot displays the Python Any Where web configuration page for the domain `heartsafe.pythonanywhere.com`. The page includes a navigation menu with tabs for 'Consoles', 'Files', 'Web', 'Schedule', and 'Databases'. The 'Web' tab is active, showing configuration options for the selected domain. Key features include a 'Reload' button, a 'Best before date' section with a 'Run until 3 months from today' button, and a 'Traffic' section with a table of site activity.

Traffic:		
How busy is your site?		
This month (previous month)	54334	(36528)
Today (yesterday)	499	(43)

Figura 3.18: Servidor gratuito “Python Any Where” donde se configura por medio de la interfaz gráfica en la pestaña web.

3.3.1 Adquisición de datos.

La captura de datos del sensor de luz “pulse sensor” se realiza por el Arduino pro mini y se envía al servidor por medio de peticiones web, en donde por el método GET se los entrega al aplicativo y se guardan posteriormente en la base de datos, el dispositivo envía una trama completa, que contiene datos específicos de latitud y longitud para localizar la posición actual del paciente, adicional también envía el valor de BPM (latidos por minuto), y los valores correspondientes de

las variables que forman la fotopleletismografía, como son: el valle, el pico, un valor estable y el periodo.

Los datos que se almacenan en la base del sistema son: latitud, longitud, código del dispositivo, BPM, valor estable, valor del valle, periodo de los pulsos, fecha y hora.

3.3.2 Modelo relacional de la base de datos.

En la figura 3.19 se presenta el esquema correspondiente a la base de datos relacional, que se encuentra alojada en el servidor, que se diseña en MySQL referente al sistema de monitoreo.

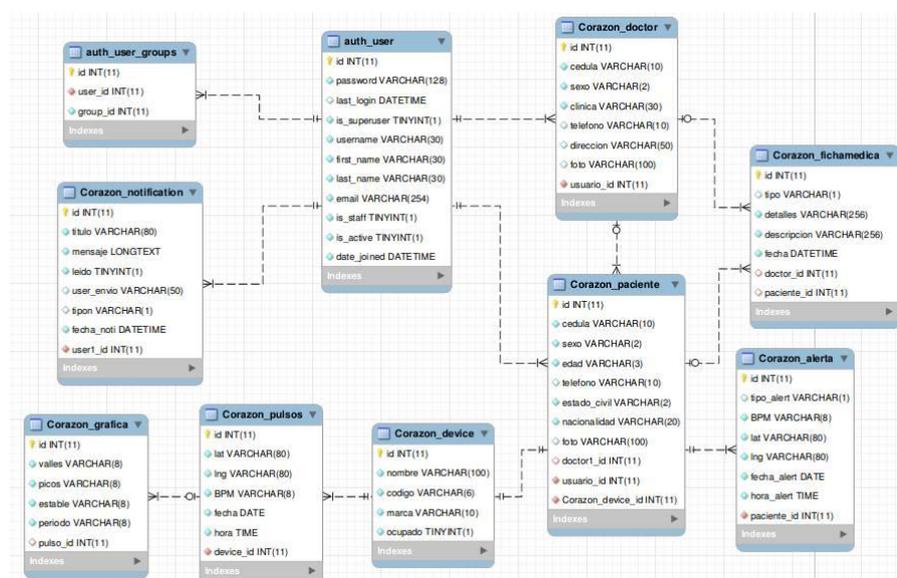


Figura 3.19: Modelo Relacional de la base de datos en MySQL.

3.4 Aplicación web.

La arquitectura web desarrollada en el sistema, explica los puntos claves en el desarrollo del aplicativo con el uso de Django, incorporando HTML, CSS, JavaScript, gestor de base de datos, servidor y el uso de las diferentes APIs, como se visualiza en la figura 3.20.



Figura 3.20: Arquitectura de desarrollo web.

La aplicación web es el medio que permite a los médicos o pacientes visualizar la información del ritmo cardíaco de un paciente específico, que se encuentra asociado a un doctor, para poder desarrollar la aplicación del sistema se trabaja con un Framework web de alto nivel llamado Django que utiliza como base la programación de Python, el cual brinda varias ventajas como la utilización de sus plantillas en la creación de una plantilla base, y la codificación de las páginas usando HTML, CSS y JavaScript.

3.4.1 Posicionamiento y Google Maps.

Para poder visualizar la posición geográfica de algún paciente se utiliza la API de Google Maps JavaScript, que se utiliza de forma gratuita, en donde se crea un mapa estableciendo como centro la última posición, el cual se actualiza cada 4 segundos, haciendo petición POST al servidor web, que contendrá las últimas coordenadas geográficas del paciente especificado, por medio de la variable global llamada "id_paciente", la cual se pasa por parámetro. Se crea un

marcador con la latitud y longitud obtenida de la petición POST, que a su vez es consultada con la última ubicación guardada en la base de datos.

Para obtener las últimas coordenadas geográficas del paciente se utiliza la función mostrada en el código que se observa en la figura 3.21.

```
function localizacionPaciente() {
    $.post( "/mapa/lngPac/"+global+"/",{ },
        function( data ) {
            // console.log("yoooo", data[i].lat, data[i].lng);

            if(entra){
                marker.setMap(null);
            }

            marker = new google.maps.Marker({
                position: new google.maps.LatLng(parseFloat(data.lat), parseFloat(data.lng)),
                map: mapglobal,
                animation: google.maps.Animation.BOUNCE,
                icon:src="/static/img/corazon_peq.png",
            });

            entra = true;

            console.log(data);
        }
    );
}
```

Figura 3.21: Función de posicionamiento de las coordenadas sobre el mapa de Google.

3.4.2 Gráficas en tiempo real.

Para el desarrollo de las dos gráficas, se utiliza la librería Highcharts, escrita en JavaScript, que corre junto con JQuery, para poder crear gráficas interactivas con gran variedad de modelos y diseños, donde se puede añadir o remover series, puntos o líneas de la gráfica luego de haberla creado. Se usa la herramienta de actualización dinámica, llamada “dynamic-update”, en donde se consulta por método POST a la ruta del servidor web, /mapa/api/grafica.dat, a la que se envía como parámetro el id del dispositivo, el cual se asigna solo uno por cada

paciente, luego se representa los puntos de la gráfica por medio de la función `addPoint`, mandando como parámetros los valores de las variables de los ejes X y Y, es importante mencionar que el eje X es con respecto al tiempo. Se realiza peticiones a la base de datos cada 10 segundos como se muestra en la figura 3.22.

```

var series = this.series[0];
setInterval(function () {
    $.post( "/mapa/api/grafica.dat",{ id_dispo:'id_dispo' },

        function( data ) {
            console.log(data); //La data se presenta en forma de diccionario
            var period = parseInt(data.periodo);
            var IBI = (period*10); //IBI=8000 mseg

            var x = (new Date()).getTime(),
                y = parseInt(data),
                x1 = Math.round((new Date()).getTime()+250),
                x2 = Math.round((new Date()).getTime()+500),
                x3 = Math.round((new Date()).getTime()+IBI);

            //Agrego los puntos en la grafica
            series.addPoint([x, parseInt(data.valles)]);
            series.addPoint([x1, parseInt(data.picos)]);
            series.addPoint([x2, parseInt(data.estable)]);
            series.addPoint([x3, parseInt(data.estable)]);

        }
    );
}, 9000); //CADA 9 seg CONSULTA A LA BASE

```

Figura 3.22: Función que dibuja los puntos de la fotopleitismografía.

3.4.3 Envío de alertas.

Las alertas son activadas por ciertos rangos que se establecen según el valor de la variable BPM. Estos rangos son determinados mediante fórmulas establecidas por los médicos. [33] Los rangos del ritmo cardíaco que se han calculado son: de 60 a 100 BPM para un ritmo cardíaco en reposo, menor a 30 BPM para un ritmo cardíaco mínimo, mayor 170 BPM para un ritmo cardíaco máximo, y menor a 150 BPM para un ritmo cardíaco de seguridad, por lo que existen tres tipos de alertas como se especifica en la figura 3.23, las alertas son enviadas

al doctor encargado y al representante del paciente por medio de correo electrónico y mensajes de texto.

INTERVALO DE ALARMAS

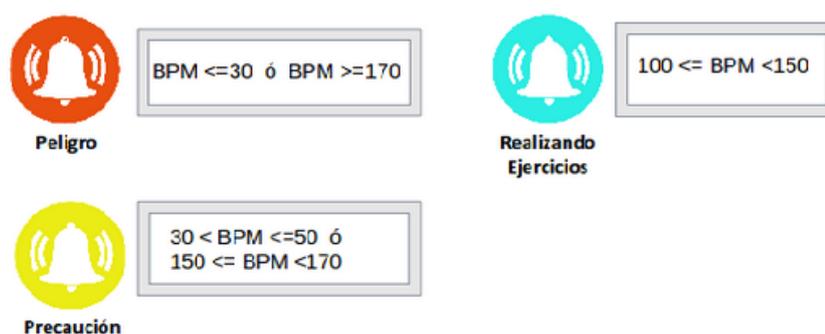


Figura 3.23: Rango de alarmas para notificar al médico y al representante del paciente ante cualquier anomalía.

Envío de correos electrónicos.

Para realizar el envío de correos electrónicos, se crea una cuenta en Gmail, para que a través de la misma se envíen las diferentes notificaciones o correos electrónicos configurando el archivo `setting.py`, donde se configura el protocolo SMTP en el puerto 587, y también se especifica el correo y contraseña de la cuenta Gmail como se aprecia en la figura 3.24.

```
#CORREO#
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_HOST_USER = 'doctor.cardiologia2016@gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_PASSWORD = 'password'
```

Figura 3.24: Archivo de configuración `setting.py` para realizar el envío de Correos Electrónicos.

Se envía la alerta de correo electrónico siempre y cuando exista un correo del paciente. Un ejemplo del formato del correo electrónico que se envía, es el siguiente:

“El paciente Martha Dávila García tiene un BPM 23. COMUNICARSE URGENTE CON EL DOCTOR”, y como asunto “Alerta”, como se puede ver en la figura 3.25.

```
#ALERTA
from django.core.mail import send_mail, BadHeaderError
if (int(pulso) <= 30 or int(pulso) >= 170):
    a = Alerta()
    a.tipo_alert = 'P'
    a.paciente = paciente
    a.BPM = pulso
    a.lat = latitud
    a.lng = longitud
    a.save()

    if (mail_destinatario != ""):
        try:
            send_mail("Alerta", "El paciente: " + nombre_completo + " tiene u
            'from@server.com', [mail_destinatario], fail_silently=False, )
        except BadHeaderError:
            return HttpResponse('Argumentos invalidos encontrados')
```

Figura 3.25: Código de la función que realiza el envío de Correos electrónicos.

Envío de mensajes de texto.

Para las alertas que se envían por SMS o mensaje de texto, se utiliza el API de Twilio, donde se crea una cuenta en la página twilio.com, que es una página que brinda varios servicios de envío de SMS, voz y notificaciones por medio de la web, se requiere usar sus propias librerías y tener una cuenta activa. Para él envío se utilizó el código que se muestra en la figura 3.26.

```
from twilio.rest import TwilioRestClient
# put your own credentials here
ACCOUNT_SID = '<AccountSid>'
AUTH_TOKEN = '<AuthToken>'

client = TwilioRestClient(ACCOUNT_SID, AUTH_TOKEN)

client.messages.create(
    to = '<ToNumber>',
    from_ = '<FromNumber>',
    body = '<BodyText>',
)
```

Figura 3.26: Código de la función que realiza el envío de mensajes de texto.

3.5 Usuarios de la aplicación web

Dentro del sistema existen 3 tipos de usuarios que pueden acceder a la aplicación web, los cuales tendrán diferentes permisos para los diversos casos de uso con los que cuenta la aplicación, el usuario principal es el administrador del sistema, quien podrá crear, editar y eliminar a los otros tipos de usuarios, el segundo tipo de usuario es el medico quien dará seguimiento a todos sus pacientes asignados por el administrador, el tercer tipo de usuario es el paciente quien también podrá monitorear el estado de su corazón o consultar alguna duda a su respectivo médico ya sea por un mensaje interno en el sistema o por correo electrónico.

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS.

En este capítulo se realiza un análisis detallado de los resultados, en el ámbito técnico como en el análisis de costos de implementación del sistema. Estas dos áreas de análisis se las realiza tanto para el dispositivo como para la aplicación web, por lo que se especifica cada resultado que se obtiene luego del desarrollo de este nuevo sistema de telemedicina. Los puntos de análisis se los detalla en la figura 4.1.

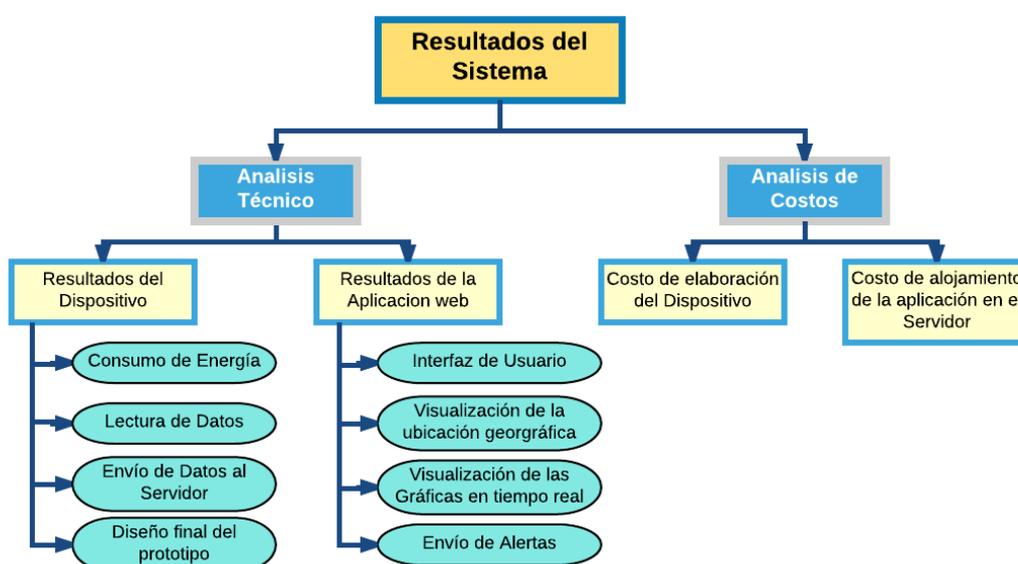


Figura 4.1: Diagrama de flujo de resultados del sistema.

A continuación, se describen detalladamente el análisis de los resultados de cada una de las áreas que se muestran en la imagen de arriba.

4.1 Análisis técnico.

En esta parte del capítulo, se describe el análisis de los resultados técnicos que se obtienen del sistema, que incluye el estudio del rendimiento del dispositivo, como también de la aplicación web, los cuales se explican a continuación.

4.1.1 Resultados del Dispositivo.

Para realizar el análisis de los resultados que involucran al dispositivo, se lo debe hacer sobre cuatro áreas importantes, que son: Consumo de energía, Lectura de datos, Envío de datos al servidor y Diseño final del prototipo.

Cada uno de los resultados referentes a estos cuatro aspectos se detallan a continuación.

Consumo de Energía.

Como se mencionó en los capítulos anteriores, el Dispositivo que realiza la lectura del ritmo cardíaco y el envío de datos al servidor web, funciona con una sola batería LiPo que suministra un voltaje nominal de 3.7 voltios, y 4.2 voltios cuando se encuentra completamente cargada. Además, la batería trabaja con una carga de 2000mAh, lo que significa que durante una hora podrá entregar constantemente una corriente de 2000mA, si el dispositivo consume una corriente menor a la indicada la duración de la batería será mayor. Por lo tanto, se debe analizar cuanto es el consumo total de corriente en el dispositivo.

Mediante el uso de un multímetro se obtiene dos valores de corriente diferentes, consumidas por el dispositivo, la primera corriente igual a 50mA, cuando el dispositivo se encuentra realizando la lectura de los datos a través del sensor de pulso y la antena Gps, y otra corriente mayor de 230 mA, cuando el dispositivo envía los datos adquiridos hacia el servidor web, ambas corrientes tienen periodos equivalentes de duración, iguales a 5 segundos.

Al usar una batería LiPo con un voltaje máximo de 4.2 y una carga de 2000mAh, y sabiendo que el dispositivo funciona a un voltaje igual al de la batería, con dos corrientes periódicas de consumo, se obtiene el tiempo de duración de la batería con los siguientes cálculos.

Primero se determina el tiempo de duración para la primera corriente que se obtiene, mediante la ecuación 4.1.

$$t1 = \frac{\text{Carga de la batería}}{I1} \quad (4.1)$$

$$t1 = \frac{2000\text{mAh}}{50\text{mA}} = 40 \text{ horas}$$

Luego se calcula el valor del tiempo para la segunda corriente, por medio de la ecuación 4.2.

$$t2 = \frac{\text{Carga de la batería}}{I2} \quad (4.2)$$

$$t2 = \frac{2000\text{mAh}}{230\text{mA}} = 8,70 \text{ horas}$$

Después se obtiene el valor del tiempo en forma ideal usando la ecuación 4.3.

$$\text{tiempo ideal} = \frac{t1+t2}{2} \quad (4.3)$$

$$\text{tiempo ideal} = \frac{40 + 8.70}{2} = 24,35 \text{ horas}$$

tiempo ideal = 24 horas y 21 minutos

Debido a que no es recomendable descargar la batería a más del 70%, para calcular el tiempo real de duración de la batería en el dispositivo se utiliza la ecuación 4.4.

$$\text{tiempo real} = (\text{tiempo ideal}) * 0,7 \quad (4.4)$$

$$\text{tiempo real} = (24,35) * (0,7) = 17,045 \text{ horas}$$

tiempo real = 17 horas y 3 minutos

Lectura de Datos

En el capítulo tres se describe, el funcionamiento del sensor a través del código del programa que se guarda en la placa Arduino, el sensor de pulso, cada dos milisegundos, realiza la lectura del ritmo cardíaco y de todas las variables que forman la fotopleletismografía.

Luego de realizar por 10 veces consecutivas, la lectura del tiempo entre cada latido o también llamado IBI en el código, calcula el valor promedio de los últimos diez valores de IBIs. Por lo tanto, la adquisición de los datos finales dentro del programa se la ejecuta cada 20 milisegundos.

Con el valor promedio del IBI, se obtiene mediante una fórmula que se aplica dentro del código, la cantidad de latidos por minuto o BPM, y todas las demás variables que forman la fotopleletismografía, como se detalla en la ecuación 4.5.

$$BPM = \frac{60000}{IBI} \quad (4.5)$$

Al colocar el sensor de pulso en la yema del dedo índice de la mano derecha de varias personas, se han obtenido valores de BPM dentro del rango normal que se establece por los médicos, que va de los 60 BPM a los 100 BPM, cuando las personas se encuentran en un estado de reposo. Cuando las mismas personas se encuentran realizando actividad física su ritmo cardíaco aumenta, obteniéndose valores menores a 155 BPM.

Cuando el dispositivo se enciende y luego realiza la primera adquisición de los datos referentes a la ubicación geográfica del paciente, se observa que la antena toma un tiempo máximo de 60 segundos en poder detectar estas coordenadas

Envío de Datos al Servidor.

Cuando el dispositivo ha obtenido a través de los sensores todos los valores que se necesitan entregar al servidor web, se ejecuta la función de envío establecida en el programa, entonces en ese momento el módulo Sim808 empieza a realizar el trabajo por el cual se lo utiliza, que es el envío de los datos al servidor web.

El envío de los datos a la web es la segunda parte más importante del dispositivo, es la que completa el proceso de adquisición y entrega de datos, que es la función que cumple el dispositivo. el envío de datos a la web se lo realiza a través de la tecnología 2G, específicamente GPRS, lo que involucra el uso de megas para realizar la conexión del dispositivo al internet.

Se determinó que la cantidad de megas por cada envío de datos al servidor web es menor a 0.001 MB. El dispositivo realiza el envío de datos a la web cada 10 segundos, el cálculo del consumo total de megas en un mes se detalla en la tabla 2.

Período de Tiempo	Número de envíos	MB consumidos
10 segundos	1 envío	0,001 MB
1 minuto	6 envíos	0,006 MB
1 hora	360 envíos	0,360 MB
1 día	8640 envíos	8,640 MB
1 mes	267840 envíos	267,840 MB

Tabla 2: Cantidad de MB por períodos de tiempo en envíos de datos al servidor web.

En el Ecuador existen 4 compañías de telefonía celular que brindan el servicio de internet, siendo la comunicación GPRS la de mayor cobertura a nivel nacional. Entre ellas la de más bajo costo para el consumidor la empresa nacional CNT, pero también se debe destacar

que, a pesar que Claro tiene los costos más altos en el país, es la compañía con mayor cobertura a nivel nacional. Por lo que la adquisición de un plan de datos para el funcionamiento del dispositivo en cualquier compañía, queda a elección del paciente.

Diseño final del prototipo.

Para poder obtener el prototipo final del dispositivo, específicamente el diseño final de su carcasa, se ha tenido que realizar tres versiones del diseño, las cuales han sido impresas para poder verificar las fallas que se obtienen luego del proceso de impresión.

La primera impresión es una versión totalmente básica de la carcasa, en la que solo se destaca su forma, tanto en la base como en el cobertor, sin definir detalles como la posición de los leds o ranuras para la colocación de la correa q se ajusta a la muñeca del brazo de la persona.

En la segunda versión se agregan los detalles antes mencionados, y también marcas referentes a la carrera, la facultad y el nombre de la institución, al momento de realizar esta impresión se visualizan algunas fallas en ciertas partes del cobertor, por lo que se procede a realizar una tercera y última versión.

En la tercera versión de la carcasa, se corrigen los errores del cobertor, aumentando el espesor de los sectores donde se localizan las fallas, obteniendo una impresión completamente buena y sin errores, como se aprecia en la figura 4.2.

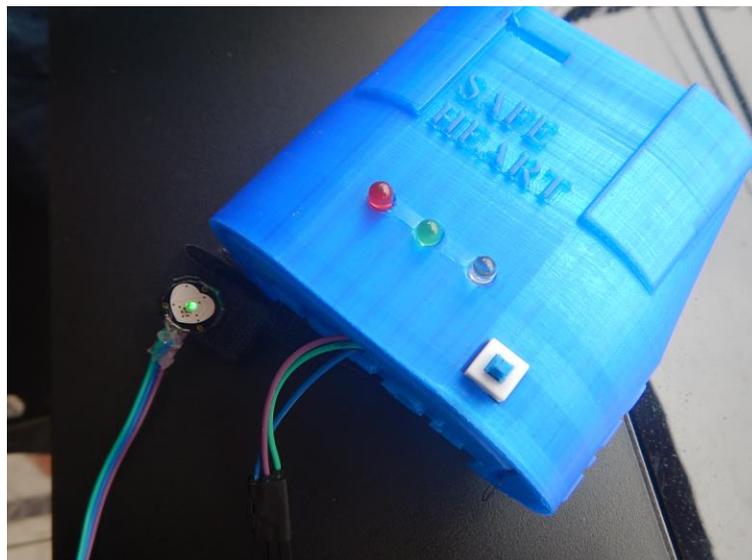


Figura 4.2: Prototipo final de la carcasa.

4.1.2 Resultados de la aplicación web.

Para realizar el análisis de los resultados referentes a la aplicación web, se lo hace en cuatro aspectos importantes, que son: interfaz de usuario, visualización de la ubicación geográfica, visualización de las gráficas en tiempo real y envío de alertas.

Cada uno de los resultados que corresponden a estas cuatro áreas se detallan a continuación.

Interfaz de Usuario.

En el capítulo anterior se menciona que existen tres tipos de usuarios que pueden ingresar a la aplicación del servidor y que cada uno cuenta con distintos privilegios dentro del sistema, por lo tanto, dentro de la aplicación existen tres interfaces de usuario distintas debido a los privilegios de cada uno de ellos.

La primera interfaz corresponde al usuario principal del sistema, que es el administrador, el cual puede registrar, editar y eliminar médicos y pacientes dentro del sistema, y también puede asignar a cada paciente registrado, un médico encargado. En las figuras 4.3 y 4.4 se

pueden visualizar las páginas de registro de médicos, y pacientes respectivamente.

SAFE HEART

- HOME
- PACIENTES
- MI PERFIL
- NOTIFICACIONES
- ADMINISTRADOR

Registrar Doctor

NOMBRE DE USUARIO
Requerido. 30 caracteres o menos. Letras, dígitos y @/./+/_ solamente.

CONTRASEÑA

CONTRASEÑA (CONFIRMACIÓN)
Para verificar, introduzca la misma contraseña anterior.

NOMBRE

APELLIDO

CÉDULA

SEXO

CLÍNICA

TELÉFONO

DIRECCIÓN

Figura 4.3: Página de registro de médicos.

SAFE HEART

- HOME
- PACIENTES
- MI PERFIL
- NOTIFICACIONES
- ADMINISTRADOR

Registrar Paciente

NOMBRE DE USUARIO
Requerido. 30 caracteres o menos. Letras, dígitos y @/./+/_ solamente.

CONTRASEÑA

CONTRASEÑA (CONFIRMACIÓN)
Para verificar, introduzca la misma contraseña anterior.

NOMBRE

APELLIDO

CÉDULA

SEXO

EDAD

ESTADO CIVIL

TELÉFONO

CORREO ELECTRONICO

Figura 4.4: Página de registro de Pacientes.

Las interfaces que se observan en las figuras 4.5 y 4.6 corresponden al segundo usuario del sistema, que es el médico, el cual puede monitorear en tiempo real la ubicación del paciente, la fotopleletismografía y la gráfica referente al valor del BPM con respecto al tiempo de cada uno de los pacientes que le han sido asignados dentro del sistema, y también puede recibir alertas con respecto al estado de sus pacientes.

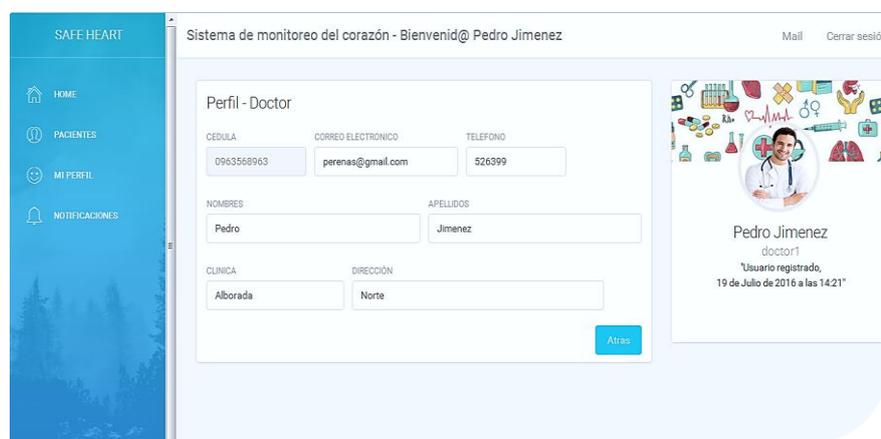


Figura 4.5: Página correspondiente a la interfaz de usuario del médico encargado.



Figura 4.6: Página que muestra la lista de pacientes según el médico que haya iniciado sesión.

La tercera interfaz que se aprecia en la figura 4.7 corresponde al tercer usuario del sistema, que es el Paciente, el cual solo puede recibir notificaciones y mensajes internos en el sistema, provenientes de su

médico encargado y del administrador del sistema como se muestra en la siguiente imagen.

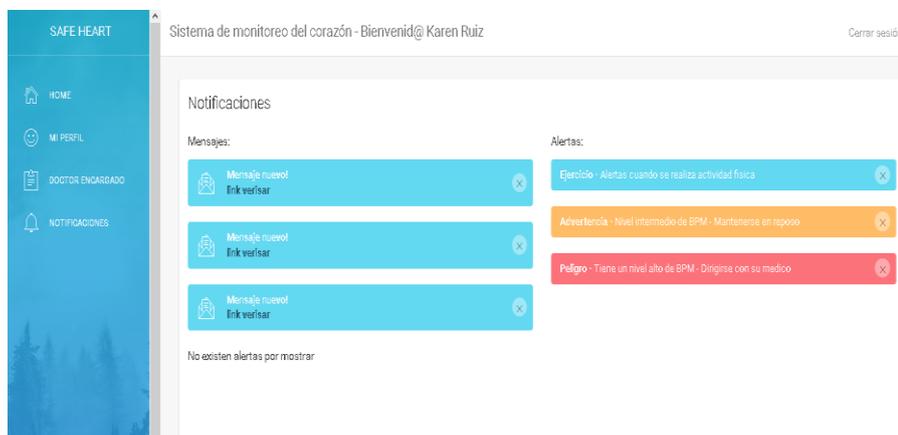


Figura 4.7: Página que muestra las principales alertas ya sea de peligro, advertencia, si esta en alguna actividad física o si ha llegado un mensaje nuevo.

Visualización de la ubicación geográfica.

Cada paciente tendrá en su dispositivo una antena GPS, por lo que podrá ser monitoreado en cada momento el lugar donde se encuentra, a través del mapa de google el cual ha sido vinculado a la aplicación, esta ubicación se actualiza cada 10 segundos. Un ejemplo de la visualización de ubicación geográfica se visualiza en la figura 4.8.



Figura 4.8: Página en donde se visualiza el mapa con la ubicación actual del paciente.

En la figura anterior se puede apreciar la página con el menú del lado izquierdo y el mapa con el respectivo marcador mostrando la ubicación actual del paciente.

Visualización de las gráficas en tiempo real.

Como el objetivo principal del sistema es monitorear el ritmo cardíaco del paciente en tiempo real, la mejor forma de hacerlo es mediante la visualización de sus respectivas gráficas, que sirven de gran ayuda al momento que el medico pueda dar un análisis.

En la figura 4.9 que se muestra a continuación, se observa la página mediante el uso de Highcharts, en la que se muestra la gráfica de PPG (Fotopletoxiografía) de un paciente específico en tiempo real.



Figura 4.9: Página en donde se visualiza la señal PPG (Fotopletismógrafo) en tiempo real.

Envío de alertas.

Una de las ventajas principales del sistema es poder alertar rápidamente de alguna anomalía del paciente respecto a su ritmo cardíaco, y así poder reducir el tiempo de reacción para atender una emergencia. Dentro del sistema existen dos formas para enviar alertas al médico, que son por mensajes de texto o por correos electrónicos.

En la figura 4.10, se muestra un ejemplo de correo electrónico enviado al médico encargado.



Figura 4.10: Envío de correo electrónico al detectar una alerta del sistema.

4.2 Análisis de costos.

En esta parte del capítulo, se describe el análisis del valor de los costos que se generan para el desarrollo del sistema, que incluye el costo de elaboración del dispositivo, como también el costo de alojamiento de la aplicación en el servidor web, los cuales se explican a continuación.

4.2.1 Costo de elaboración del dispositivo.

Para la implementación del dispositivo se debe adquirir algunos elementos electrónicos, como la placa de arduino, el módulo Sim808, el sensor de pulso, la antena Gps, la antena Gsm / Gprs y la batería, cada uno de estos elementos tienen un precio de venta en el mercado, a estos valores se debe sumar el costo de impresión 3D, para de esta forma tener un precio estimado en la elaboración del dispositivo.

Cada uno de los valores referentes a los elementos adquiridos y de impresión se los detalla en la tabla 3.

Descripción del producto	Precio
Arduino Pro mini 8Mhz	\$ 1,80
Módulo Sim808 con GPS, GSM/GPRS	\$ 34,03
Pulse Sensor	\$ 3,23
Antena GPS pasiva con conector UFL	\$ 3,25
Antena GSM/GPRS con conector UFL	\$ 1,99
Batería LiPo 3.7v de una celda 2000mAh	\$ 3,92
Impresión de carcasa en 3D	\$ 50,00
TOTAL:	\$ 98,22

Tabla 3: Costos en la implementación del dispositivo.

Cabe recalcar que el precio de los productos descritos en la tabla exceptuando el valor de la impresión en 3D, son valores en el mercado internacional, en países como China y Estados Unidos, como se puede ver los costos fuera del país son mucho más económicos que dentro, en el Ecuador estos valores pueden duplicarse o muchas veces hasta triplicarse. La impresión en 3D se la realiza dentro del

país, por lo que su costo es el único a nivel nacional, dando como resultado un valor de \$98,22 de elaboración del dispositivo, un valor económico respecto a los beneficios que brinda.

4.2.2 Costo del servidor web.

Para completar la implementación del sistema se necesita alojar nuestra aplicación en un servidor web que brinde los beneficios que se requiere. Se ha utilizado un servidor gratuito llamado “PythonAnyWhere”, para aplicaciones desarrolladas en Python como la de este sistema. El costo de uso del servidor es totalmente gratuito durante los primeros tres meses de uso, luego de esto el servicio tiene un costo de \$5,00 por cada mes, lo que es un precio económico considerando el alcance y número de usuarios que pueden registrarse en el sistema. [34]

CONCLUSIONES Y RECOMENDACIONES

En conclusión, se obtiene un sistema de telemedicina que ayuda al médico a monitorear en tiempo real el ritmo cardíaco del paciente, la cantidad de latidos por minutos y su ubicación actual, lo cual ayuda a detectar cualquier tipo de anomalía y atenderla en un tiempo corto.

Se determinó el medio más conveniente de lectura del ritmo cardíaco para el funcionamiento del sistema, a través de un fotosensor, llamado "Pulse Sensor", que es un fotopleletismógrafo por reflexión de luz, diferenciándose básicamente en sus fuentes de adquisición de datos. El sensor solo requiere contacto con una parte del cuerpo, por medio de la fotocelda capta la intensidad de luz que rebota en el hueso del dedo, y que debido al paso de la sangre esta luz puede ser de mayor o menor intensidad.

Se brinda un módulo de lectura del ritmo cardíaco, mediante el uso del sensor indicado en el párrafo anterior, que permite la obtención de datos analizando los periodos de lectura óptimos a configurar.

Se logra la integración de hardware y software de manera que se monitoree la frecuencia cardíaca de una persona a través de un sensor y visualice en un aplicativo web; este aplicativo permite geolocalización, y observación de datos mediante gráficos, reportes, para poder ser analizado por un médico o usuario pertinente.

Se estableció la comunicación por medio de conexión remota usando tecnología 2G específicamente GPRS, basado en un análisis comparativo de cobertura a nivel nacional con tecnologías 3G y 4G, y el tamaño de los datos que se obtienen en el dispositivo que son transmitidos hacia el servidor web.

Se creó un prototipo de bajo costo del dispositivo, de alto rendimiento que puede ser portable en la muñeca del brazo de una persona, capaz de integrar los componentes electrónicos en su interior y aportar al desarrollo tecnológico orientado a IoT.

Se recomienda que el sensor sea usado en el dedo índice de la mano derecha del paciente.

El sensor puede ser colocado con cinta plegable o velcro de color negro, de esta manera se logra disminuir en gran proporción la pérdida de luz que genera el led del sensor.

Se recomienda crear un circuito electrónico propio, en los que se integre el controlador y el módulo Sim808 en la misma placa, para poder disminuir las dimensiones del dispositivo.

Se puede agregar al dispositivo un oxímetro de pulso, que ayuda a medir la cantidad de oxígeno en la sangre, y de esta forma aportar una variable más al análisis médico del paciente.

El sistema puede ser complementado, desarrollando y agregándole una aplicación móvil, que permita visualizar el ritmo cardíaco de un paciente a través de una gráfica y también pueda recibir alertas, a través de un teléfono celular.

BIBLIOGRAFÍA

- [1] OMS. (2015, Enero 13). Organización Mundial de la Salud. [Online]. Disponible en: www.who.int/mediacentre/factsheets/fs317/es/. [Último acceso: 15 Abril 2016].
- [2] Organización Mundial de la Salud. (2015, Septiembre). Preguntas y respuestas sobre la hipertensión. [Online]. Disponible en: <http://www.who.int/features/qa/82/es/>. [Último acceso: 10 Junio 2016].
- [3] M. J. Ackerman, C. A. Marcou y D. J. Teste, «Medicina personalizada: diagnóstico genético de cardiopatías/canalopatías hereditarias,» *Revista Española de Cardiología*, vol. 66, pp. 298-307, 2013.
- [4] R. S. Istepanian, *The Potential of Internet of Things (IOT) for Assisted Living Applications*, London: The Institution of Engineering and Technology, 2011.
- [5] N. Bari, G. Mani y S. Berkovich, *Internet of Things as a Methodological Concept*, Washington DC, USA, 2013, pp. 1-4.
- [6] G. Yang, L. Xie, M. Mäntysalo, X. Zhou y Z. Pang, «A Health-IoT Platform Based on the Integration of Intelligent Packaging, Unobtrusive Bio-Sensor, and Intelligent Medicine Box,» *IEEE*, vol. 10, nº 4, pp. 1-4, NOVEMBER 2014.
- [7] W. Uribe Arango, M. Duque Ramírez y E. Medina Durango, «Electrocardiografías y Arritmias,» Bogota, P.L.A. Export Editores Ltda., 2007, pp. 28-39.
- [8] INEC. (2015, Junio 15). Ecuador en cifras. [Online]. Disponible en: http://www.ecuadorencifras.gob.ec/documentos/web-inec/Poblacion_y_Demografia/Nacimientos_Defunciones/2014/. [Último acceso: 15 Abril 2016].
- [9] L. Higuera Ortega. (2015, Febrero). Electrocardiograma ECG/EKG. Fundacion Española del Corazón. [Online]. Disponible en:

- <http://www.fundaciondelcorazon.com/informacion-para-pacientes/metodos-diagnosticos/electrocardiograma.html>. [Último acceso: 27 junio 2016].
- [10] World Famous Electronics llc. (2016). Open Hardware. [Online]. Disponible en: <http://pulsesensor.com/pages/open-hardware>. [Último acceso: 20 junio 2016].
- [11] J. C. C. Mateu. (2012, Octubre 17). Análisis del electrocardiograma (ECG). [Online]. Disponible en: <http://pfcjccaimarimateu.blogspot.com/2012/10/analisis-del-electrocardiograma-ecg.html>. [Último acceso: 3 agosto 2016].
- [12] G. Celi, M. Rocha y M. Yapur, *mediciones fotopletimográficas.pdf*, Guayaquil, Guayas, 2011.
- [13] T. Ward. (2015). Department of Electronic Engineering The Photoplethysmograph as an instrument for physiological measurement Tomás Ward Department of Electronic Engineering. [Online]. Disponible en: <http://slideplayer.com/slide/4417979/>. [Último acceso: 12 julio 2016].
- [14] SIMCom Wireless Solutions Co., Ltd. (2015). SIM808. [Online]. Disponible en: <http://simcomm2m.com/En/module/detail.aspx?id=137>. [Último acceso: 20 06 2016].
- [15] Arduino. (2015). Arduino Pro Mini. [Online]. Disponible en: <https://www.arduino.cc/en/Main/ArduinoBoardProMini>. [Último acceso: 20 junio 2016].
- [16] icstation. (2015-2016). 3.7V Lithium Polymer Battery 103450 2000mAh PCM Rechargeable. [Online]. Disponible en: <http://www.icstation.com/lithium-polymer-battery-103450-2000mah-rechargeable-p-7399.html>. [Último acceso: 28 junio 2016].
- [17] L. Giraldo. (2015, Marzo 20). Technolugi e Informática. [Online]. Disponible en: http://technolugi.blogspot.com/2015_03_01_archive.html. [Último acceso: 19 junio 2016].

- [18] Federación de Enseñanza de CC.OO. de Andalucía. (2010, Enero). La conectividad inalámbrica. [Online]. Disponible en: <https://www.feandalucia.ccoo.es/docu/p5sd6785.pdf>. [Último acceso: 21 julio 2016].
- [19] SGS GROUP. (2013). 1991datasheet. [Online]. Disponible en: <https://cdn-shop.adafruit.com/datasheets/1991datasheet.pdf>. [Último acceso: 20 junio 2016].
- [20] Electronica Embajadores. (2016). ANTENA GPS PASIVA uFL - 2 dBi. [Online]. Disponible en: <http://www.electronicaembajadores.com/Productos/Detalle/1/AT1A010/antena-gps-pasiva-ufl---2-dbi>. [Último acceso: 21 junio 2016].
- [21] A. Mocholí. (2015, Septiembre 15). 6 Tipos de desarrollo de aplicaciones web. yeePLY. [Online]. Disponible en: <https://www.yeePLY.com/blog/6-tipos-desarrollo-de-aplicaciones-web/>. [Último acceso: 16 julio 2016].
- [22] SM. (2015, Septiembre 7). Arduino Software (IDE). ARDUINO. [Online]. Disponible en: <https://www.arduino.cc/en/Guide/Environment#>. [Último acceso: 21 junio 2016].
- [23] Django Software Foundation. Django makes it easier to build better Web apps more quickly and with less code. [Online]. Disponible en: <https://www.djangoproject.com/>. [Último acceso: 28 junio 2016].
- [24] Python Software Foundation. Welcome to Python.org. [Online]. Disponible en: <https://www.python.org/>. [Último acceso: 22 junio 2016].
- [25] W3Schools. (1999). HTML Introduction. [Online]. Disponible en: http://www.w3schools.com/html/html_intro.asp. [Último acceso: 25 junio 2016].
- [26] Documentation Group. (2016, Abril 11). Welcome! - The Apache HTTP Server Project. [Online]. Disponible en: <https://httpd.apache.org/>. [Último acceso: 26 junio 2016].

- [27] Oracle Corporation and/or its affiliates. (2016). MySQL. [Online]. Disponible en: <https://www.mysql.com/>. [Último acceso: 26 junio 2016].
- [28] Autodesk, Inc. (2016). Inventor | Software de diseño mecánico y CAD 3D | Autodesk. [Online]. Disponible en: <http://www.autodesk.es/products/inventor/overview>. [Último acceso: 27 junio 2016].
- [29] Lucid Software Inc. (2016). Flowchart Maker & Online Diagram Software. [Online]. Disponible en: <https://www.lucidchart.com/pages/b/cta>. [Último acceso: 27 junio 2016].
- [30] F. B. Lletí, *Comunicación Bluetooth entre Arduino UNO y Android aplicado a un detector de mentiras*”, GANDIA, 2015.
- [31] World Famous Electronics llc. Pulse Sensor Amped. [Online]. Disponible en: <http://pulsesensor.com/pages/pulse-sensor-amped-arduino-v1dot1>. [Último acceso: 20 Julio 2016].
- [32] PythonAnywhere LLP. (2016). Host, run, and code Python in the cloud!. [Online]. Disponible en: <https://www.pythonanywhere.com/>. [Último acceso: 3 julio 2016].
- [33] Polar Electro. (2005, septiembre 2). Preguntas Frecuentes. [Online]. Disponible en: http://polarcr.com/html/preguntas/ritmo_cardiaco.html. [Último acceso: 3 julio 2016].
- [34] PythonAnywhere LLP. (2016). Plans and pricing : PythonAnywhere. [Online]. Disponible en: <https://www.pythonanywhere.com/pricing/>. [Último acceso: 5 julio 2016].
- [35] Emiliano, (2007, Diciembre 7). Universo celular. [Online]. Disponible en: <http://universocelular.com/2007/12/07/gsm-y-gprs-conceptos-generales/>. [Último acceso: 15 Abril 2016].

- [36] M. Rouse. (2007, Mayo). Search mobile computing. [Online]. Disponible en: <http://searchmobilecomputing.techtarget.com/definition/GSM>. [Último acceso: 15 Abril 2016].
- [37] S. Blardone. (2013, Noviembre 17). Telemedicina: ¿qué es y para qué se utiliza?. infobae. [Online]. Disponible en: <http://www.infobae.com/2013/11/17/1524294-telemedicina-que-es-y-que-se-utiliza/>. [Último acceso: 2 septiembre 2016].
- [38] K. Conway y M. DeHaan. (2011, Febrero). What is UART (Universal Asynchronous Receiver/Transmitter)? - Definition from Whatls.com. Whatls.com. [Online]. Disponible en: <http://whatis.techtarget.com/definition/UART-Universal-Asynchronous-Receiver-Transmitter>. [Último acceso: 3 agosto 2016].
- [39] J. Sánchez. (2006, Septiembre 29). ¿Qué es un 'framework'?. 2006. [Online]. Disponible en: <http://jordisan.net/blog/2006/que-es-un-framework/>. [Último acceso: 5 agosto 2016].
- [40] M. Rouse. (2005). SMTP (Simple Mail Transfer Protocol). techtarget. [Online]. Disponible en: <http://searchexchange.techtarget.com/definition/SMTP>. [Último acceso: 15 agosto 2016].

ANEXOS

ANEXO A – GLOSARIO.

GPS: Es un sistema que permite determinar en todo el mundo la posición de un objeto con una alta precisión.

SMS: Es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos, conocidos como mensajes de texto.

GPRS: General Packet Radio Service es una tecnología de conmutación de paquetes, que permite la transferencia de datos a través de redes celulares. Se utiliza para el Internet móvil, MMS y otras comunicaciones de datos. [35]

GSM: (Sistema Global para Comunicaciones Móviles) es un sistema de telefonía móvil digital que se utiliza ampliamente en Europa y otras partes del mundo. GSM digitaliza y comprime los datos, a continuación, envía a través de un canal con otras dos corrientes de datos de usuario, cada uno en su propia ranura de tiempo. Opera en cualquiera de los 900 MHz o 1800 MHz banda de frecuencia. [36]

MB: Representa la simbología recomendada para el Megabyte, que es un múltiplo del byte, igual a 1000000 de bytes de información.

UFL: Es un tipo de conector coaxial, con una frecuencia de gran rendimiento de 6 GHz, fabricada con bronce fosforado, sirve para conectar dispositivos de comunicación inalámbrica como antenas de teléfonos, antenas Gps o cualquier función que precise de alta frecuencia de transmisión.

LiPo: Representa la abreviatura para las baterías hechas de polímero de iones de litio.

API: Abreviatura en inglés para la interfaz de programación de aplicaciones (Application Programming Interface), contiene las funciones y procedimientos almacenadas en las librerías, para que puedan ser usadas por otros programas.

RFID: Definición de identificación por radiofrecuencia. Se refiere a dispositivos electrónicos pequeños que tienen un pequeño chip y una antena.

Telemetría: Tecnología de medición remota de magnitudes físicas para ser transmitidas a un observador lejano en el sistema.

Telemedicina: La telemedicina es un sistema que ofrece servicios de medicina a distancia, utilizando tecnologías de información y comunicaciones. [37]

UART: (Universal Asynchronous Receiver/Transmitter). Es el microchip con la programación que controla la interfaz de un ordenador a sus dispositivos conectados en serie. De manera que pueda intercambiar datos con módems y otros dispositivos en serie. [38]

TTL: (Transistor-Transistor Logic), es una tecnología de fabricación de circuitos digitales usando transistores bipolares que origina la lógica de transistor a transistor.

C: Es un lenguaje de programación estructurado, permite el uso de la recursividad y la creación de funciones.

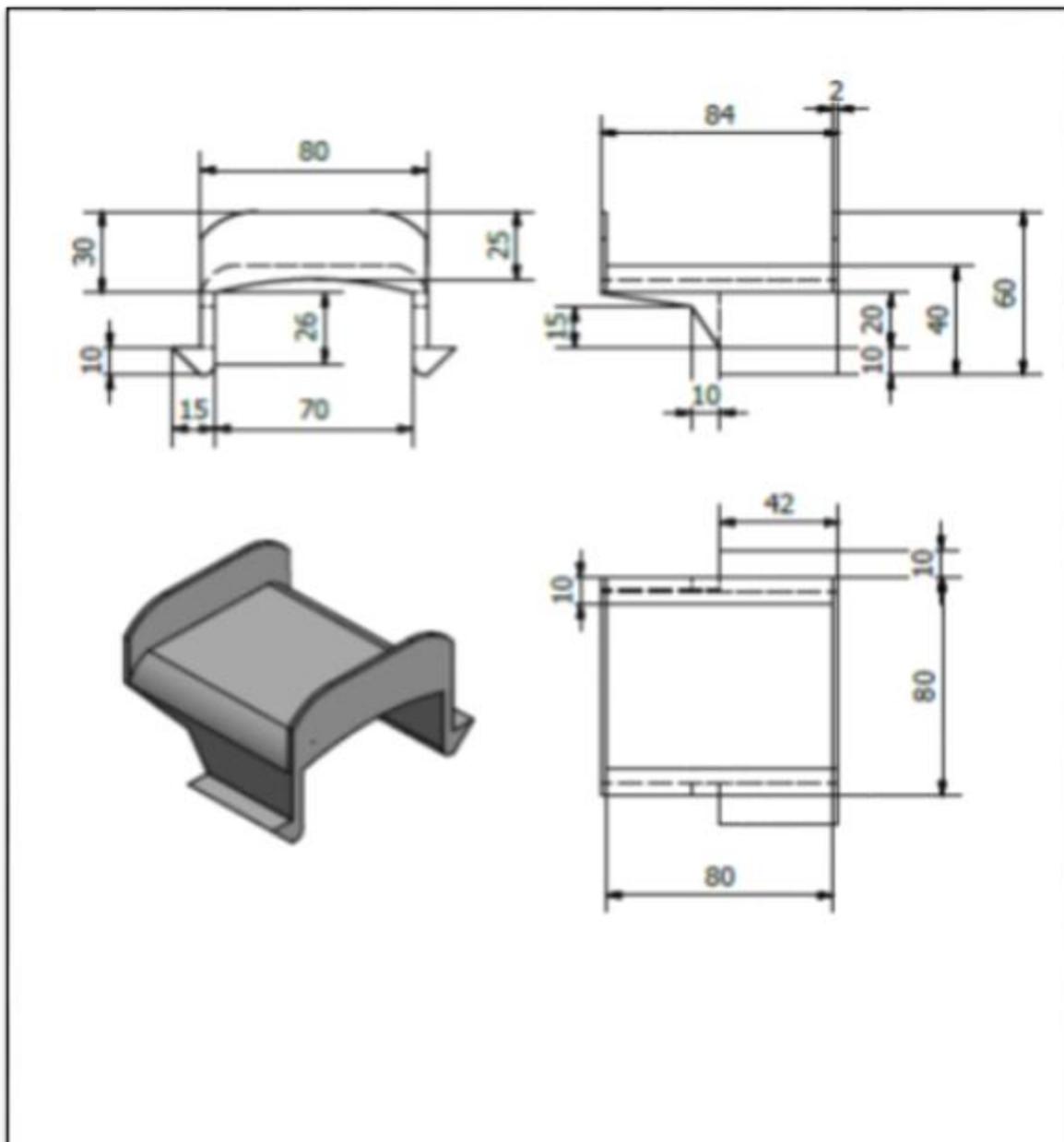
Framework: Es un esquema que sirve para el desarrollo y la implementación de una aplicación. La mayoría de ellos están ligados a un lenguaje en específico, pero esto no sucede siempre. [39]

Vcc: Es uno de los dos pines de alimentación que van conectados a la fuente de corriente del circuito donde están instalados, vienen en casi todos los circuitos integrados.

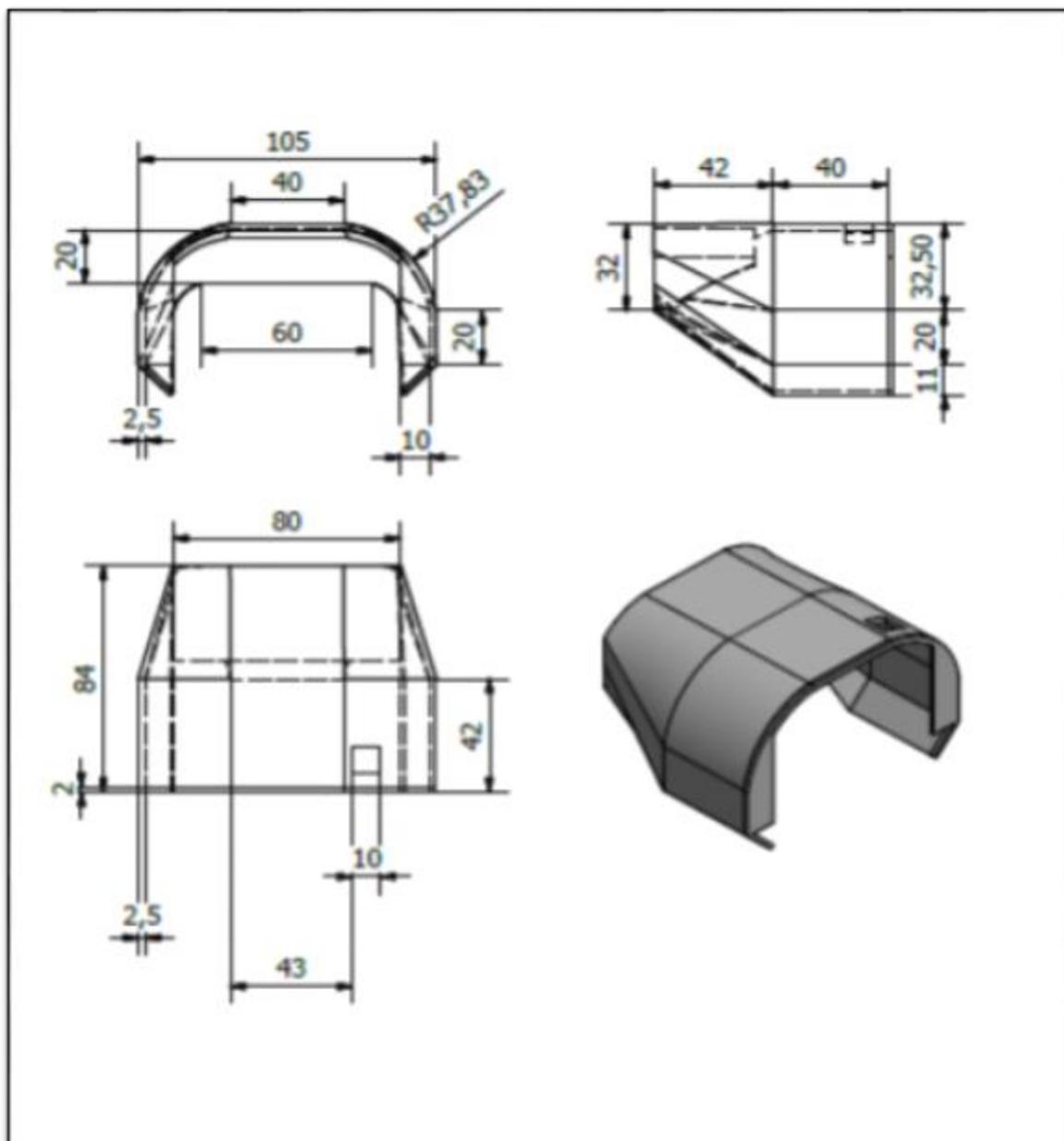
Gnd: Es el punto cero de todos los voltajes eléctricos que estén en un circuito eléctrico. línea por donde fluye la corriente, donde se cierra el circuito, es conocido como el polo negativo.

SMTP: (Simple Mail Transfer Protocol) es un protocolo TCP / IP que se utiliza en el envío y recepción de correo electrónico. [40]

ANEXO B – DISEÑO DE LA CARCASA EN 3D.



VISTA DE LA BASE CON DIMENSIONES EN MILÍMETROS.



VISTA DE LA CUBIERTA CON DIMENSIONES EN MILÍMETROS.

ANEXO C – CODIGO DEL DISPOSITIVO EN ARDUINO.

```

#include <SoftwareSerial.h>
#include <String.h>

SoftwareSerial mySerial (7, 8);

//char Doctor_cell[] = "0987213512"; // numero de telefono
//char Relative_cell[] = "0995972732"; // numero de telefono

char frame[500];
char id_dispo[] = "12345";

String lat = "";
String lon = "";

String seguir="";

String pulse;
String peak;
String valley;
String stable;
String period;

int tamano;
int tamano1;
int tamano2;
int tamano3;
int tamano4;
int tamlat;
int tamlon;
String coordenadas;

// VARIABLES
int pulsePin = A0; // Pulse Sensor purple wire connected to analog p
                    pin 0
int blinkPin = 5; // pin to blink led at each beat
int VCC = 2;
int Powerkey = 9;
int AlertPin = 4;
int WorkPin = 10;

// these variables are volatile because they are used during the
interrupt service routine!
volatile int BPM; // used to hold the pulse rate
volatile int PICO;

```

```

volatile int VALLE;
volatile int TIEMPO;
volatile int NORMAL;
volatile int Signal;           // holds the incoming raw data
volatile int IBI = 600;       // holds the time between beats,
the Inter-Beat Interval
volatile boolean Pulse = false; // true when pulse wave is high,
false when it's low
volatile boolean QS = false;  // becomes true when Arduino
finds a beat.

void setup() {
  pinMode(Powerkey, OUTPUT);
  Serial.println(" INICIA TODO ");
  power();
  pinMode(blinkPin, OUTPUT);    // pin that will blink to your
heartbeat!
  pinMode(AlertPin, OUTPUT);
  pinMode(WorkPin, OUTPUT);
  pinMode(VCC, OUTPUT);
  digitalWrite(VCC, HIGH);
  mySerial.begin(2400);         // the GPRS baud rate
  Serial.begin(19200);        // the serial baud rate
  delay(500);
  configuracion_inicial();
  Serial.println("COPMUNICACION GPRS HABILITADA");
  digitalWrite(WorkPin, LOW);
  interruptSetup();           // sets up to read Pulse Sensor
signal every 2mS
}

void loop() {
  sensar(); // function
}

void power(void)
{
  digitalWrite(Powerkey, HIGH);
  delay(3000);                // wait for 3 seconds
  digitalWrite(Powerkey, LOW);
  delay(10000);               // wait for 10 seconds
}

void configuracion_inicial()
{
  mySerial.println("AT+IPR=2400");//modo texto

```

```
delay(300);
ShowSerialData();

mySerial.println("AT+CSQ");
delay(100);
ShowSerialData();

mySerial.println("AT+CMGF=1");//modo texto
delay(2000);
seguir=mySerial.read();
if(seguir == "OK"){
    digitalWrite(WorkPin, HIGH);
}

mySerial.println("AT+CGATT?");
delay(100);
ShowSerialData();

mySerial.println("AT+SAPBR=3,1,\"CONTYPE\", \"GPRS\");//setting
the SAPBR, the connection type is using gprs
delay(1000);
ShowSerialData();

mySerial.println("AT+SAPBR=3,1,\"APN\", \"internet.movistar.com.ec\"");
//setting the APN, the second need you fill in your local apn
server
delay(4000);
ShowSerialData();

mySerial.println("AT+SAPBR=1,1");//setting the SAPBR, for detail
you can refer to the AT command manual
delay(2000);
ShowSerialData();

mySerial.println("AT+HTTPINIT"); //init the HTTP request
delay(2000);
ShowSerialData();

mySerial.println("AT+CGNSPWR=1");
delay(2000);
ShowSerialData();

mySerial.println("AT+CGNSSEQ=RMC");
delay(45000);
ShowSerialData();
}
```

```

////////////////////
// Where the Magic Happens
void sensar() {
  if (QS == true) { // A Heartbeat Was Found
    SubmitHttpRequest(BPM,PICO,VALLE,NORMAL,TIEMPO); // A Beat
    Happened, Output that to serial.
    QS = false; // reset the Quantified Self
    flag for next time
  }
  delay(20); // take a break
}

//////////Enviar mensaje de texto de alerta al Medico responsable
//void enviar_mensaje(int dato)
//{
//  String LPM=String(dato);
//  mySerial.print("AT+CMGS="); // comando de envio de mensaje
//  mySerial.print((char)34); //se pone las comillas ", para que
// lo tome debe ser char de lo contrario el serial envia caracter por
// caracter
//  mySerial.print(Doctor_cell); //se coloca número de teléfono
// del Medico
//  mySerial.println((char)34); //se vuelve a poner el carácter "
//  delay(200); //tiempo para que de respuesta el módulo
//  mySerial.print("Advertencia!! El ritmo cardíaco del paciente
// es:" +LPM+" BPM"); //texto del mensaje de texto
//  mySerial.print((char)26); // se pone el símbolo ASCII 26, que
// corresponde a CTRL+Z, con lo que el módulo sabe que el sms termino
//  LPM="";
//}

//////////ENVIO DE datos a la web.
void SubmitHttpRequest(int dato, int dato1, int dato2, int dato3,
int dato4)
{
  coordenadas = GPS();
  Serial.println(coordenadas);
  int firstFind = coordenadas.indexOf(",");
  int secondLat = coordenadas.indexOf(",", firstFind + 3);
  int thirdLon = coordenadas.indexOf(",", secondLat + 2);
  lat = coordenadas.substring(secondLat + 1, secondLat + 10);
  lon = coordenadas.substring(thirdLon + 1, thirdLon + 10);
  Serial.println(lat);
  Serial.println(lon);
  Serial.println(dato);
}

```

```

Serial.println(dato1);
Serial.println(dato2);
Serial.println(dato3);
Serial.println(dato4);

pulse = String(dato);
peak = String(dato1);
valley = String(dato2);
stable = String(dato3);
period = String(dato4);

tamlat = lat.length() + 1;
tamlon = lon.length() + 1;
tamano = pulse.length() + 1;
tamano1 = peak.length() + 1;
tamano2 = valley.length() + 1;
tamano3 = stable.length() + 1;
tamano4 = period.length() + 1;

char latitud[tamlat];
char longitud[tamlon];
char pulso[tamano];
char picos[tamano1];
char valles[tamano2];
char estable[tamano3];
char periodo[tamano4];

lat.toCharArray (latitud, tamlat);
lon.toCharArray(longitud, tamlon);
pulse.toCharArray(pulso, tamano);
peak.toCharArray(picos, tamano1);
valley.toCharArray(valles, tamano2);
stable.toCharArray(estable, tamano3);
period.toCharArray(periodo, tamano4);

mySerial.print("AT+HTTTPARA=\"URL\", \"http://heartsafe.pythonanywher
e.com/mapa/conexion2/?"); // setting the httppara, the second
parameter is the website you want to access
    sprintf(frame,
"id_dispo=%s&pulso=%s&latitud=%s&longitud=%s&picos=%s&valles=%s&esta
ble=%s&periodo=%s", id_dispo, pulso, latitud, longitud, picos,
valles, estable, periodo);
    mySerial.print(frame);
    mySerial.println("");
    delay(1000);
    ShowSerialData();

```

```

    mySerial.println("AT+HTTPACTION=0");//submit the request
    delay(7000);//the delay is very important, the delay time is base
on the return from the website, if the return datas are very large,
the time required longer.
    ShowSerialData();

    mySerial.println("AT+HTTPREAD");// read the data from the website
you access
    delay(300);
    ShowSerialData();

    //verifica si existe una alerta para mostrarla en el led de
alertas
    if((dato < 40)|| (dato > 170)){
        digitalWrite(AlertPin,HIGH);
    }
    else{
        digitalWrite(AlertPin,LOW);
    }
}

void ShowSerialData()
{
    while (mySerial.available() != 0)
        Serial.write(mySerial.read());
}

String GPS() {
    while (1) {
        mySerial.println("AT+CGNSINF");
        ShowSerialData();
        String response = "";
        long int time = millis();
        while ((time + 1000) > millis())
        {
            while (mySerial.available())
            {
                char c = mySerial.read();
                response += c;
            }
        }
        return (response);
        delay(10000);
    }
}

```

```

//Segunda página del código donde realiza la interrupción

volatile int rate [10];    // used to hold last ten IBI values
volatile unsigned long sampleCounter = 0; // used to determine pulse
timing
volatile unsigned long lastBeatTime = 0; // used to find the inter
beat interval
volatile int P =512;      // used to find peak in
pulse wave
volatile int T = 512;    // used to find trough in
pulse wave
volatile int thresh = 512; // used to find instant
moment of heart beat
volatile int amp = 100;   // used to hold amplitude
of pulse waveform
volatile boolean firstBeat = true; // used to seed rate array
so we startup with reasonable BPM
volatile boolean secondBeat = true; // used to seed rate array
so we startup with reasonable BPM

void interruptSetup(){
  // Initializes Timer2 to throw an interrupt every 2mS.
  TCCR2A = 0x02;    // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO
INTO CTC MODE
  TCCR2B = 0x05;    // DON'T FORCE COMPARE, 128 PRESCALER
  OCR2A = 0X7C;    // SET THE TOP OF THE COUNT TO 124 FOR 500Hz
SAMPLE RATE
  TIMSK2 = 0x02;    // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND
OCR2A
  sei();           // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
}

// THIS IS THE TIMER 2 INTERRUPT SERVICE ROUTINE.
// Timer 2 makes sure that we take a reading every 2 miliseconds
ISR(TIMER2_COMPA_vect){ // triggered when
Timer2 counts to 124
  cli(); // disable
interrupts while we do this
  Signal = analogRead(pulsePin); // read the Pulse
Sensor
  sampleCounter += 2; // keep track of the
time in mS with this variable
  int N = sampleCounter - lastBeatTime; // monitor the time
since the last beat to avoid noise

```

```

// find the peak and trough of the pulse wave
    if(Signal < thresh && N > (IBI/5)*3){ // avoid dichrotic
noise by waiting 3/5 of last IBI
        if (Signal < T){ // T is the trough
            T = Signal; // keep track of
lowest point in pulse wave
            VALLE=T;
        }
    }

    if(Signal > thresh && Signal > P){ // thresh condition
helps avoid noise
        P = Signal; // P is the peak
        PICO=P; // keep track of
highest point in pulse wave
    }

    // NOW IT'S TIME TO LOOK FOR THE HEART BEAT
    // signal surges up in value every time there is a pulse
    if (N > 250){ // avoid high
frequency noise
        if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3) ){
            Pulse = true; // set the Pulse
flag when we think there is a pulse
            digitalWrite(blinkPin,HIGH); // turn on pin 13
LED
            IBI = sampleCounter - lastBeatTime; // measure time
between beats in mS
            lastBeatTime = sampleCounter; // keep track of
time for next pulse

            if(firstBeat){ // if it's the first
time we found a beat, if firstBeat == TRUE
                firstBeat = false; // clear firstBeat
flag
                return; // IBI value is
unreliable so discard it
            }
            if(secondBeat){ // if this is the
second beat, if secondBeat == TRUE
                secondBeat = false; // clear secondBeat
flag
                for(int i=0; i<=9; i++){ // seed the running
total to get a realisitic BPM at startup
                    rate[i] = IBI;
                }
            }

```

```

    }

    // keep a running total of the last 10 IBI values
    word runningTotal = 0; // clear the
runningTotal variable

    for(int i=0; i<=8; i++){ // shift data in the
rate array // and drop the oldest
        rate[i] = rate[i+1];
IBI value // add up the 9 oldest
        runningTotal += rate[i];
IBI values
    }

    rate[9] = IBI; // add the latest IBI to
the rate array // add the latest IBI to
    runningTotal += rate[9];
runningTotal // average the last 10
    runningTotal /= 10;
IBI values
    TIEMPO=runningTotal;
    BPM = 60000/runningTotal; // how many beats can
fit into a minute? that's BPM!
    QS = true; // set Quantified Self
flag
    // QS FLAG IS NOT CLEARED INSIDE THIS ISR
    }
}

    if (Signal < thresh && Pulse == true){ // when the values are
going down, the beat is over // turn off pin 13 LED
        digitalWrite(blinkPin,LOW); // reset the Pulse flag
        Pulse = false;
so we can do it again // get amplitude of the
        amp = P - T;
pulse wave // set thresh at 50% of
        thresh = amp/2 + T;
the amplitude
        NORMAL= amp/4 + T;
        P = thresh; // reset these for next
time
        T = thresh;
    }

    if (N > 2500){ // if 2.5 seconds go by
without a beat // set thresh default
        thresh = 512;

```

```
        P = 512; // set P default
        T = 512; // set T default
        PICO=P;
        VALLE=T;
        NORMAL=thresh;
        TIEMPO=2500;
        lastBeatTime = sampleCounter; // bring the
lastBeatTime up to date
        firstBeat = true; // set these to avoid
noise
        secondBeat = true; // when we get the
heartbeat back
    }

    sei(); // enable interrupts
when youre done!
} // end isr
```