UNIVERISDAD DEL AZUAY

FACULTAD DE CIENCIA Y TECNOLOGÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA

Sistema de telemetría para adquisición y procesamiento de bioseñales para neonatos.

Trabajo de graduación previo a la obtención del título de:
INGENIERO ELECTRÓNICO.

Autores:

PATRICIO GABRIEL OVIEDO RIERA.

PAUL ANDRÉS VALDIVIESO MORA.

Director:

OMAR SANTIAGO ALVARADO CANDO.

CUENCA, ECUADOR
2016

DEDICATORIA

A nuestras familias y amigos, quienes siempre nos han motivado y apoyado, de forma incondicional, durante nuestra formación personal y académica.

Gabriel Oviedo - Paul Valdivieso

AGRADECIMIENTOS

A todos los profesores, quienes compartieron sus conocimientos y contribuyeron a nuestra formación profesional y que, además, aportaron y apoyaron a la idea que dio origen a este proyecto.

Gabriel Oviedo - Paul Valdivieso

ÍNDICE DE CONTENIDOS

DEDICATORIA ii

AGRADECIMIENTOS iii

ÍNDICE DE CONTENIDOS iv

ÍNDICE DE FIGURAS viii

ÍNDICE DE TABLAS x

ÍNDICE DE ANEXOS xi

RESUMEN xii

ABSTRACT xiii

INTRODUCCIÓN 1

CAPÍTULO 1: MARCO TEORICO 3

1.1. Cunas de calor radiante 3

1.1.1 Antecedentes 3

1.1.2 Principio de funcionamiento de cunas de calor radiante 4

1.1.3 Partes de una cuna de calor radiante 5

1.1.4 Tipos de cunas de calor radiante 6

1.1.4.1 Cunas de calor radiante con control manual 7

1.1.4.2 Cunas de calor radiante servo controladas 7

1.1.5 Normativas 7

1.2. La Telemedicina 8

1.2.1 Antecedentes 8

1.2.2 Tipos de Telemedicina 9

1.2.3 Telemedicina en la neonatología 11

CAPÍTULO 2: ADQUISICIÓN Y TRANSMISIÓN DE BIOSEÑALES 12

2.1. Hardware 12

2.1.1 Plataforma biomédica e-Health V2.0 12

2.1.1.1 Funciones generales de la plataforma e-Health 14

2.1.2 Arduino Uno 16

2.1.3 Raspberry Pi 2 model B 18

2.2. Lectura de Sensores 19

2.2.1 Balanza 20

2.2.1.1 Celdas de carga 20

2.2.1.2 Ajuste del puente 22

2.2.1.3 Acondicionamiento de la señal 23

2.2.1.4 Filtrado 23

2.2.1.5 Fuente de alimentación 24

2.2.2 Control de temperatura 25

2.2.2.1 Circuito de potencia 26

2.2.3 Circuito ECG 26

2.2.3.1 Filtro ECG 28

2.2.4 Sensor pulsioxímetro 28

2.3. Procesamiento de datos 29

2.3.1 Procesamiento Arduino 30

2.3.1.1 Arduino 1 30

2.3.1.2 Arduino 2 31

2.3.2 Procesamiento en Raspberry 32

2.3.2.1 Pantalla login 33

2.3.2.2 Pantalla datos 34

2.3.2.3 Pantalla monitor 36

CAPÍTULO 3: SERVIDOR WEB Y CLIENTE 38

3.1. Servidor WEB 38

3.1.1 Lenguaje de Programación 38

3.1.2 Protocolo de Comunicación XML-RPC 38

3.1.3 NetBeans 39

3.1.4 Diseño Servidor 39

3.1.4.1 Clase Servidor 41

3.1.4.2 Clase Usuario 42

3.1.4.3 Clase Paciente 42

3.1.4.4 Clase DatoPaciente 42

3.1.4.5 Clase DatoSignos 42

3.1.4.6 Clase Alta 43

3.2. Aplicación de Web 43

3.2.1 Java Server Faces 43

3.2.2 PrimeFaces 43

3.2.3 HTML 44

3.2.4 Diseño Pagina Web 44

3.2.4.1 Página Inicial 47

3.2.4.2 Página de signos 48

3.3. Base de datos 49

3.3.1 MySQL 49

3.3.2 Descripción Base de Datos 50

3.4. Desarrollo de aplicación para Clientes 50

3.4.1 Android Studio 50

3.4.2 Diseño de Aplicación 51

3.4.2.1 Pantalla y Clase Login 51

3.4.2.2 Pantalla y Clase Signos 53

3.4.2.3 Pantalla y Clase ECG 55

CAPÍTULO 4: ANÁLISIS Y RESULTADOS 57

4.1. Instrumentación Biomédica 57

4.2. Procesadores 58

4.3. Servidor y clientes 59

4.4. Estructura 59

CONCLUSIONES Y RECOMENDACIONES 64

BIBLIOGRAFÍA 66

ANEXOS 70

ÍNDICE DE FIGURAS

Figura 1.1: Incubadoras desarrolladas por Lyon en 1981. 4
Figura 1.2: Principio de funcionamiento de una cuna de calor radiante. 5
Figura 1.3: Partes de una cuna de calor radiante. 6
Figura 1.4: Tipos de Telemedicina. 9
Figura 2.1: Tipos de sensores compatibles con e-Health V2.0. 13
Figura 2.2: Distribución de sensores en plataforma e-Health cara superior. 13
Figura 2.3: Distribución sensores en plataforma e-Health cara posterior. 14
Figura 2.4: Raspberry Pi 2 model B. 18
Figura 2.5: Montaje de la tarjeta e-Health. 20
Figura 2.6: Configuración interna de una celda de carga. 20
Figura 2.7: Celda de carga SEN-10245. 21
Figura 2.8: Conexionado de celdas de carga. 22
Figura 2.9: Circuito de ajuste para puente de Wheatstone. 22
Figura 2.10: Configuración buffer. 23
Figura 2.11: Amplificación con OPAM de instrumentación INA131AP. 23
Figura 2.12: Filtro pasa bajo Butterworth de cuarto orden. 24
Figura 2.13: Circuito inversor para fuente simétrica. 24
Figura 2.14: Circuito de temperatura. 25
Figura 2.15: Control de potencia para calefactor. 26
Figura 2.16: Circuito de acondicionamiento de ECG. 27
Figura 2.17: Filtro para ECG. 28
Figura 2.18: Circuito de captura para pulsioxímetro 29
Figura 2.19: Diagrama del sistema. 29
Figura 2.20: Diagrama de flujo Arduino1. 31

Figura 2.21: Diagrama de flujo Arduino 2. 32
Figura 2.22: Sistema de procesamiento Raspberry. 33
Figura 2.23: Pantalla login. 33
Figura 2.24: Diagrama de flujo para pantalla login. 34
Figura 2.25: Pantalla datos. 35
Figura 2.26: Diagrama pantalla datos. 35
Figura 2.27: Pantalla monitor. 36
Figura 2.28: Diagrama de flujo pantalla monitor. 37
Figura 3.1: Diagrama Servidor. 41
Figura 3.2: Diagrama Página de inicio. 45
Figura 3.3: Diagrama página de signos. 46
Figura 3.4: Pagina Inicial. 47
Figura 3.5: Mensaje de error login. 48
Figura 3.6: Página Web de Signos vitales. 48
Figura 3.7: Pantalla Login para aplicación Android. 52
Figura 3.8: Notificaciones de error en pantalla Login. 53
Figura 3.9: Diagrama Pantalla signos para clientes Android. 54
Figura 3.10: Pantalla de signos vitales para clientes Android. 55
Figura 3.11: Diagrama pantalla ECG para clientes Android. 56
Figura 3.12: Grafica ECG en para clientes Android. 56
Figura 4.1: Filtrado de la señal ECG. 58
Figura 4.2: Base de la estructura. 60
Figura 4.3: Columna principal con soporte para cuna. 61
Figura 4.4: Soporte y cuna. 62
Figura 4.5: Estructura pintada. 62
Figura 4.6: Estructura finalizada. 63

ÍNDICE DE TABLAS

Tabla 1.1: Normas de una cuna de calor radiante. 8

Tabla 2.1: Funciones de librería e-Health. 15

Tabla 2.2: Especificaciones técnicas del Arduino Uno. 17

Tabla 2.3: Características especiales de los pines. 17

Tabla 2.4: Especificaciones Raspberry pi 2. 19

Tabla 2.5: Características de celda de carga SEN-10245. 21

ÍNDICE DE ANEXOS

Anexo 1: Circuito de balanza 70

Anexo 2: Circuito de filtrado para ECG 71

Anexo 3: Circuito de potencia para control de carga 72

Anexo 4: Código para configuración PID 73

Anexo 5: Código de configuración para Arduino 1 74

SISTEMA DE TELEMETRIA PARA ADQUISICIÓN Y PROCESAMIENTO DE BIOSEÑALES PARA NEONATOS

RESUMEN

El proyecto propuesto es un sistema de telemetría de bioseñales para neonatos capaz de distribuir información oportuna y en tiempo real sobre el estado de un paciente, al personal médico, dentro de la red LAN hospitalaria. Para este propósito, se utiliza hardware existente que dispone de sensores y la instrumentación necesaria para obtener varios parámetros fisiológicos, los mismos que son interpretados por un microprocesador para ser visualizados en una pantalla y transmitidos, de forma simultánea, hacia un servidor web donde son almacenados junto con la información del paciente para poder ser consultados a través de la aplicación cliente. Debido a la cantidad elevada de información manipulada, la eficiencia del sistema depende de la capacidad de procesamiento de los dispositivos empleados.

Palabras Clave: Electrocardiografía, Pulsioximetria, Telemedicina, Servocuna, Servicio WEB, Android.

Omar Santiago Alvarado Cando.
Director de Titulación.

Hugo Marcelo Torres Salamea.
Director de Escuela.

Patricio Gabriel Oviedo Riera.

Paúl Andrés Valdivieso Mora.

Autores

TELEMETRY SYSTEM FOR ACQUISITION AND PROCESSING OF BIO SIGNALS FOR NEW BORN INFANTS

ABSTRACT

The proposed project is about a bio signal telemetry system for neonates capable of delivering to the medical staff real-time timely information of a patient's status, within the hospital LAN (Local Area Network). For this purpose, the existing hardware with sensors and instrumentation to obtain various physiological parameters is used; which is interpreted by a microprocessor so as to be displayed on a screen and transmitted simultaneously to a web server, where they are stored along with patient information in order to be accessed through client's application. Due to the high amount of information handled, the efficiency of the system depends on the processing capacity of the devices used.

Keywords: Electrocardiography, Pulse Oximetry, Telemedicine, *Servocuna*, WEB Service, Android.

Omar Santiago Alvarado Cando
Thesis Director

Hugo Marcelo Torres Salamea
School Director

Patricio Gabriel Oviedo Riera

Paúl Andrés Valdivieso Mora

Authors

UNIVERSIDAD DEL AZUAY
Dpto. Idiomas

Translated by,
Lic. Lourdes Crespo

Patricio Gabriel Oviedo Riera.
Paul Andrés Valdivieso Mora.
Trabajo de Titulación.
Ing. Omar Santiago Alvarado Cando.
Junio, 2016.

SISTEMA DE TELEMETRIA PARA ADQUISICIÓN Y PROCESAMIENTO DE BIOSEÑALES PARA NEONATOS

INTRODUCCIÓN

En la actualidad, el avance tecnológico crece rápida y constantemente, haciendo de ella una herramienta útil y eficaz en diversos aspectos de la vida humana, y, entre sus aplicaciones se puede resaltar el cuidado de la salud. En base a esto, La medicina ha hecho de la tecnología una de sus herramientas más preciadas, facilitando procesos que antes eran imposibles de realizar y poniendo al campo de la electrónica y otras disciplinas como pilares fundamentales en el desarrollo de instrumentación y control de procesos médicos.

Además, las prestaciones que ofrecen las telecomunicaciones han permitido integrar servicios para el cuidado de la salud mediante el uso de dispositivos móviles ofreciendo una atención más personalizada, sin embargo, la mayoría de estos sistemas se enfocan principalmente al ámbito deportivo lo que genera una brecha en el cuidado de paciente de alto riesgo, como los recién nacidos prematuros.

Es por eso que el proyecto aquí planteado trata de poner en práctica los conocimientos adquiridos en las asignaturas de redes de computadoras, programación java y Android así como también los conceptos de bio-electrónica para desarrollar un software que sea capaz de adquirir ciertos parámetros fisiológicos, tales como: saturación de oxígeno, frecuencia cardíaca, temperatura corporal y peso

de un recién nacido, los cuales son obtenidos a través de sensores que van conectados a una plataforma biomédica que está acoplada a un prototipo de servo-cuna. Los datos obtenidos son interpretados y posteriormente enviados a un servidor web, el cual muestra la información al personal médico que realice la consulta de un paciente, cabe resaltar que todo el intercambio de información se realiza dentro de la red LAN hospitalaria.

Para la adquisición de las bioseñales se emplea la tarjeta biomédica e-Health, la cual posee la instrumentación básica para la adquisición de varios parámetros fisiológicos mediante el uso de sensores, la tarjeta va montada sobre la plataforma Arduino, donde se lee y digitaliza las señales mediante un conversor analógico/digital de diez bits, además, las prepara para ser transmitidas, en un orden especifico, hacia el procesador central (Raspberry Pi) empleando la comunicación serial (USB).

El Raspberry Pi hace la interpretación de los datos recibidos y separa la información, de cada parámetro, para ser visualizados en una pantalla y simultáneamente ser transmitidos, a través del protocolo Ethernet, hacia un servidor WEB dentro de la red LAN. Todo el proceso lo realiza una aplicación programada en el lenguaje Python que, además, permite comparar las señales y verificar que estén dentro de los rangos programados permitidos, de no ser así, se genera una alarma sonora y visual que indica lo que está sucediendo.

El servidor WEB no es más que un equipo informático que está al servicio de otros dispositivos, llamados clientes, y que gestiona una aplicación programada que, para este caso particular, está programada en el lenguaje JAVA, la cual recibe la información y la almacena en una base de datos para posteriormente ser consumida por los clientes por medio de una página WEB o de una aplicación Android, donde el personal puede ver reflejado el estado del paciente.

CAPÍTULO 1

MARCO TEORICO

Resumen: Este capítulo presenta una breve introducción sobre el origen y principio de funcionamiento de la termo cuna, la cual sirve como soporte vital para los recién nacidos prematuros ya que, debido a la susceptibilidad de su organismo, necesitan de un medio controlado que simule en cierto grado las condiciones del vientre materno, también se aborda el tema de la contribución de la tecnología que en conjunto con la medicina han generado diversos tipos de servicios que contribuyen al cuidado de la salud de las personas, así como también, la incursión de dicha tecnología al área de la neonatología.

1.1. Cunas de calor radiante

1.1.1 Antecedentes

El cuidado a los recién nacidos prematuros, antes del siglo XIX, era escaso, o nulo, y su recuperación se dejaba principalmente a su evolución, lo que producía un alto índice de mortalidad infantil, debido a esto, era indispensable desarrollar alternativas para mejorar la atención de los infantes recién nacidos, siendo las parteras y los obstetras franceses los primeros en generar aportaciones en el área de la neonatología. (Morilla, 2006).

Dentro de estas aportaciones, se observó que el control térmico era uno de los factores críticos que se debía tener presente, pues la muerte por hipotermia fue el principal inconveniente que debía ser superado, por lo que se han venido proponiendo métodos que proporcionen calor al neonato, lo cual también sería el primer paso para dar origen a las primeras termo cunas (Egan, 2006). Uno de los primeros intentos por mantener el calor del recién nacido consistía en llenar botellas de vidrio con agua caliente, lo que evitaba el enfriamiento durante el traslado. Otro método fue el de regular una flama, de forma manual, la cual calentaba el aire de la sala de maternidad proporcionando calor al neonato. (Pelaez, 2014).

Por otra parte, fue en el año de 1878 donde el obstetra S. Tarnier en conjunto con su colaborador, Odile Martinse, desarrollaron un artefacto para evitar la mortalidad en niños prematuros, cuyo peso era inferior a los dos kilogramos, basándose en el principio de una incubadora para la cría de pollos. Más adelante, en el año 1891, Alexander Lyons también realizó algunas mejoras y lanzó al mercado su incubadora, pero, fue debido a Pierre Budín, considerado por muchos como el padre de la neonatología, quien señalo en su libro *Le Nourrisson*, de 1907, que para garantizar la supervivencia de los neonatos prematuros se debía controlar la temperatura, lo que impulsó el perfeccionamiento de las incubadoras de la época, las cuales posteriormente serían introducidas a Inglaterra, Estados Unidos y Cuba, donde se generaría un desarrollo acelerado de las mismas (Morilla, 2006) hasta llegar a lo que hoy en día vemos y que en conjunto con otros equipos, es capaz de medir la actividad cardíaca, presión sanguínea, saturación de oxígeno en la sangre, entre otras características.

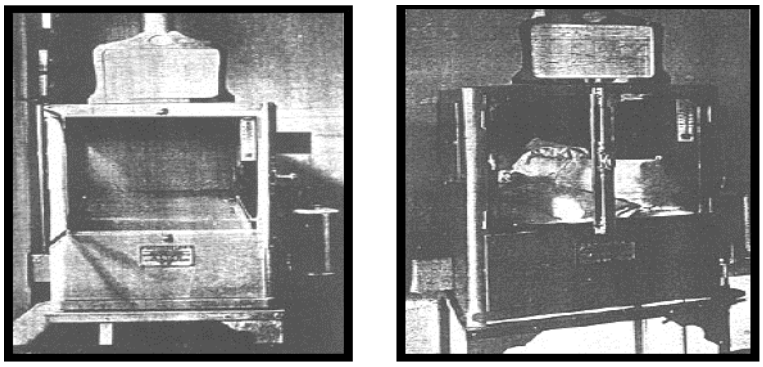


Figura 1.1: Incubadoras desarrolladas por Lyon en 1981.
Fuente: (Morilla, 2006).

1.1.2 Principio de funcionamiento de cunas de calor radiante

Las cunas radiantes son dispositivos creados para dar calor constante a los recién nacidos de bajo peso que presentan problemas de termorregulación o que requieren de intervenciones constantes por parte del personal médico, su propósito es el de mantener una zona termo neutral entre los 36ºC y 37ºC donde el consumo de oxígeno y el metabolismo del recién nacido se reducen al mínimo, permitiendo que los nutriente ingeridos se dediquen al desarrollo y crecimiento de su organismo. Para poder generar una zona termo neutral se hace uso de la transmisión de calor, desde

una fuente hacia el paciente, haciendo uso de cualquiera de las tres formas de transmisión de la energía calorífica, siendo estas: conducción, convección y radiación. (Pelaez, 2014) (Mora, RUELAS, & VELAZQUEZ, 2004).

Es así que las cunas de calor radiante, como su nombre lo indica, hacen uso de transferencia de energía por radiación para transmitir el calor en forma de ondas electromagnéticas que viajan a través del aire, aunque, existen dispositivos que también hacen uso de la transferencia de calor por convección para ayudar a calentar el colchón donde reposa el recién nacido. En este tipo de dispositivos, la fuente de calor está separada del receptor (el neonato) por lo menos a un metro de distancia y debido a que a no generan un ambiente cerrado no es necesario humedecer el aire como en el caso de las incubadoras. (Pelaez, 2014) (Mora, RUELAS, & VELAZQUEZ, 2004).

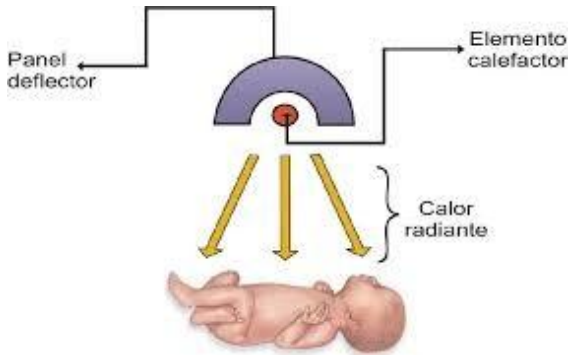


Figura 1.2: Principio de funcionamiento de una cuna de calor radiante.
Fuente: (Mora, RUELAS, & VELAZQUEZ, 2004).

1.1.3 Partes de una cuna de calor radiante

Las cunas de calor radiante están constituidas por tres bloques principales, como indica la Figura 1.3. (Pelaez, 2014) (Mora, RUELAS, & VELAZQUEZ, 2004).

- Bloque uno: está constituido por la fuente de calor, que puede ser de varios tipos:
  - Tubos de cuarzo, cerámicos o de luz infrarroja, difusores,
  - Lámparas incandescentes, etc.
- Bloque dos: es la unidad de control que incluye:

- Alarmas audibles y visibles predeterminas de fábrica o ajustables por el operador.
- Control manual del calefactor,
- Control automático del calefactor o servo controlado.

- Bloque tres: está constituido por una plataforma o base sobre la cual se encuentra un colchón, paredes transparentes que puede ser o no abatibles, con canaletas para sujeción de venoclisis, tubos de ventilación, sensores, transductores, etc. Porta chasis para placas de rayos X.

Figura 1.3: Partes de una cuna de calor radiante.
Fuente: (Mora, RUELAS, & VELAZQUEZ, 2004).

1.1.4 Tipos de cunas de calor radiante

Las cunas de calor radiante, debido al sistema de control de la calefacción, se pueden dividir en dos grupos:

1.1.4.1 Cunas de calor radiante con control manual

En este tipo de cunas, la variable a controlar es la cantidad máxima de calor que debe entregar el calefactor, siendo este un valor establecido por el operador. Este sistema requiere una supervisión constate por parte del personal médico a cargo, ya que de no realizar una supervisión adecuada se presenta el riesgo de causar quemaduras debido a la hipertermia. A fin de evitar esta situación, la mayoría de fabricantes incluyen una alarma, la cual recuerda al operador que realice una revisión de la temperatura del paciente. (Pelaez, 2014) (Mora, RUELAS, & VELAZQUEZ, 2004).

1.1.4.2 Cunas de calor radiante servo controladas

A diferencia de las cunas con control manual, la variable a controlar es la temperatura corporal del neonato. Este sistema tiene dos tipos de control para el funcionamiento del calefactor, el primero alterna el estado del calefactor de encendido a apagado de forma similar al de las cunas con control manual, siendo un sistema de precaución en caso de que fallen los sensores, el segundo consiste en regular progresivamente el calor emitido por el calefactor en función de los cambio de temperatura del bebé, estos cambios son detectados por sensores que van adheridos a la piel del paciente. El propósito de este sistema es mantener la temperatura del bebe igual a la temperatura establecida por el operador (zona termo neutral). (Pelaez, 2014) (Mora, RUELAS, & VELAZQUEZ, 2004).

1.1.5 Normativas

En la Tabla 1.1 se listan las nomas principales que deben cumplir las cunas de calor radiante.

Tabla 1.1: Normas de una cuna de calor radiante.

Norma	Expedida por	Año
NOM-137-SSA 1-1995, Información gratulatoria especificaciones generales de etiquetado que deberán ostentar los dispositivos médicos, tanto de manufactura nacional como procedencia	Secretaria de salud, México	1998
NOM-001-SEDE-1999, Instalaciones eléctricas.	Secretaria de energía, México	1999
601221-RC01. Medical electrical equipment – part 2: particular requirements for the safety of infant radian warmers.	ANSI/AAMI	2000
BS 5724: section 2.25. Specification for servo-controlled impact radian warmers.	BS	1988
IEC 60601-1 (1988-12). 1988. Medical electrical equipment –part 1: general requirements for	IEC	1988
IEC 60601-1-am1 (1991-11). Medical electrical equipment –part 1: general requirements for safety. Amendment 1.	IEC	1988
IEC 60601-1-am2 (1995-03). Medical electrical equipment –part 1: general requirements for safety. Amendment 2.	IEC	1995
IEC 60601-1-2 (1993-04). Medical electrical equipment –part 1: general requirements for safety. Section 2. Collateral standard: electromagnetic compatibility –requirements and	IEC	1993
IEC 60601-2-50 (2000-07). Medical electrical equipment –part 2: particular requirements for the safety of infant phototherapy equipment.	IEC	2000

Fuente: (Mora, RUELAS, & VELAZQUEZ, 2004).

1.2. La Telemedicina

1.2.1 Antecedentes

El uso de las tecnologías de la información y comunicación también ha venido desempeñando un rol importante dentro de la medicina, una de sus primeras incursiones data del año de 1910, realizado por Einthoven, donde hace uso de la líneas análogas de telefonía para la transmisión de electrocardiogramas 'ECG' y electroencefalogramas 'EEG', además de la implementación de la clave morse, en el año de 1920, para brindar consulta médica. El resultado de esta unión dio origen a lo que hoy se conoce como telemedicina, la cual se define, según la organización mundial de la salud 'OMS', como "el suministro de servicios de atención sanitaria, cuando la distancia representa un factor crítico, por profesionales de la salud que

apoyan el uso de las tecnologías de la información y comunicación con el objeto de intercambiar información para diagnósticos, preconizar tratamientos y prevenir enfermedades, así como como para la formación permanente de los profesionales de atención de salud a fin de mejorar la salud de las personas". (Ruiz Ibáñez, Zuluaga De Cadena, & Trujillo Zea, 2007).

En la actualidad, existen varias ramas que se han generado a partir de la cohesión entre las tecnologías de la información y comunicación con la medicina, entre las cuales sobresalen la telemedicina, la telesalud, el telecuidado y la e-salud. La telesalud se orienta a la transmisión de información que permita administrar servicios clínicos, administrativos y educativos. Por otra parte, en lo que respecta al cuidado personalizado de los pacientes, el telecuidado emplea las tecnologías de la información y comunicación para transmitir información médica válida para el diagnóstico y la terapia de pacientes desde la comodidad del hogar, mientras que, la e-salud se enfoca en obtener el conocimiento sobre las necesidades de los ciudadanos, pacientes y profesionales de la salud sobre la prestación de los diversos servicios en salud. (Ruiz Ibáñez, Zuluaga De Cadena, & Trujillo Zea, 2007).

1.2.2 Tipos de Telemedicina

Dentro de la telemedicina también existen algunas divisiones que varían en función a su alcance, de las cuales se pueden considerar cuatro principales, a continuación la Figura 1.4 indica cuales son:

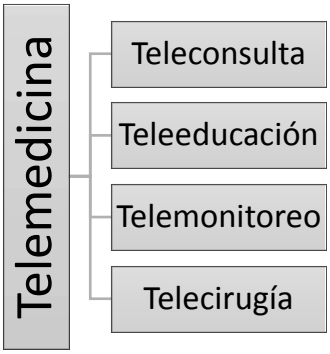


Figura 1.4: Tipos de Telemedicina.
Fuente: (Ruiz Ibáñez, Zuluaga De Cadena, & Trujillo Zea, 2007).

La teleconsulta utiliza las tecnologías de la información y comunicación para realizar búsquedas de información médica o para el asesoramiento del personal médico hacia los pacientes o entre profesionales , ya sea dentro de la localidad hospitalaria o de forma remota a la misma, la teleconsulta se puede realizar de dos maneras, síncrona o asíncrona. Cuando se envía información en la cual la revisión o asesoramiento de la misma se realiza posteriormente, se habla de una consulta asíncrona o también llamada almacenamiento y envío, por otra parte, cuando la información transmitida tiene que ser consultada en tiempo real, es decir; el asesoramiento se lo debe realizar de forma inmediata, se dice que es una consulta síncrona, este tipo de transmisión es utilizada para enviar datos que contienen valoración fisiológica como la saturación de oxígeno, presión sanguínea, electrocardiografía, entre otras. (Ruiz Ibáñez, Zuluaga De Cadena, & Trujillo Zea, 2007).

La teleeducación se enfoca en la formación educativa médica a distancia, siendo las tecnologías basadas en Internet o videoconferencias las más utilizadas, su propósito es el mejoramiento de la práctica de los conocimientos, brindando oportunidades de entrenamiento a través de consultas remotas con especialistas, adicionalmente, contribuye a reducir costos, tiempos de viaje y ausencia de personal dentro de programas de educación. (Ruiz Ibáñez, Zuluaga De Cadena, & Trujillo Zea, 2007).

El telemonitoreo utiliza las tecnologías de la información y comunicación para obtener información específica o de rutina respecto a la condición de un paciente y permite a los especialistas de la salud monitorizar las distintas variables fisiológicas, imágenes, sonidos o resultados de exámenes para determinar el tratamiento correspondiente. (Ruiz Ibáñez, Zuluaga De Cadena, & Trujillo Zea, 2007).

La telecirugía contribuye a la medicina con procedimientos quirúrgicos donde el cirujano no actúa en las cercanías del paciente, debido a la inaccesibilidad, ya sea por ambientes hostiles o por factores que comprometan la integridad del equipo médico o del paciente, impidiendo que sean atendidos dentro de los estándares convencionales, por esta razón la manipulación de los instrumentos y la visión se realiza a través de equipos electrónicos de alta tecnología. (Ruiz Ibáñez, Zuluaga De Cadena, & Trujillo Zea, 2007).

1.2.3 Telemedicina en la neonatología

Con el avance de la tecnología y la creciente demanda de las comunicaciones móviles, se han creado dispositivos portátiles como pulseras, cinturones e incluso textiles dotados de sensores, que registran las actividades fisiológicas del cuerpo, pero, están enfocados principalmente al deporte y a las áreas recreativas, es por eso que no resultan útiles para ser empleados en pacientes de alto riesgo, como los recién nacidos prematuros, debido a la simplicidad de los mismos, los cuales proveen pocos parámetros del paciente y con un análisis en línea muy escaso o nulo (Anliker, y otros, 2004), haciendo que la mayoría de la información recolectada se efectúe de forma manual y se almacene en hojas clínicas, estas anotaciones dependerán del personal médico a cargo que lo escriba y que generalmente se realiza en intervalos de 60 minutos o, en el mejor de los casos, en intervalos de 30 minutos, por eso es aconsejable que para los recién nacidos prematuros, cuyos parámetros varían constantemente, el registro de información también se realice minuto a minuto y se lo almacene en un documento electrónico. (Eklund, McGregor, & Smith, 2008).

Cuando se trata a neonatos prematuros, es importante tener consideraciones especiales debido que todo nacimiento prematuro, menor a las 37 semanas de gestación, altera drásticamente las condiciones normales del neonato respecto al entorno intrauterino (Ciani, y otros, 2008) e impide el desarrollo completo de su organismo (Eklund, McGregor, & Smith, 2008) volviéndolos vulnerables a las perturbaciones externas, es por eso que, el monitoreo constante en la unidad de cuidados intensivos neonatales ‘UCIN’ de un hospital es crucial para la detección de eventos anómalos o posibles complicaciones. (Chen, Nguyen, Coops, Oetomo, & Feijs, 2009).

Varias de las investigaciones enfocadas al soporte de las unidades de cuidados intensivos se concentran principalmente en la generación de alarmas clínicas, esta información generalmente es limitada y solo contiene un pequeño grupo de parámetros fisiológicos o información clínica del paciente (Eklund, McGregor, & Smith, 2008), y a pesar que los hospitales están dotados de equipos capaces de centralizar la información esta posteriormente se pierde o no llega al personal médico en forma oportuna.

CAPÍTULO 2

ADQUISICIÓN Y TRANSMISIÓN DE BIOSEÑALES

Resumen: Éste capítulo presenta la información más relevante sobre los dispositivos utilizados para la adquisición de las bioseñales, así como también, la metodología para el procesamiento, visualización y envío de la información hacia el servidor.

2.1. Hardware

2.1.1 Plataforma biomédica e-Health V2.0

La plataforma biomédica e-Health V2.0 es una tarjeta de propósito general, desarrollado por la empresa española Cooking Hacks, que permite a los usuarios de Arduino y Raspberry Pi realizar aplicaciones biomédicas para monitorizar, en tiempo real, el estado de un paciente o para la adquisición de datos críticos que pueden ser analizados posteriormente para la realización de un diagnóstico médico. La información recolectada, dependiendo del tipo de aplicación, puede ser enviada de forma inalámbrica utilizando cualquiera de los siguientes tipos de transmisión: WI-FI, 3G, GPRS, Bluetooth, 802.15.4 y Zig Bee, además, de ser necesario se puede enviar imagen y vídeo utilizando una cámara que se acopla a un módulo 3G compatible con la plataforma. (Hacks, 2013).

Para la adquisición de señales biomédicas, la plataforma dispone de varios sensores que permiten obtener diez aspectos fisiológicos distintos, los cuales son: frecuencia cardíaca, saturación de oxígeno en la sangre ‘SPO2’, flujo de aire o frecuencia respiratoria, temperatura corporal, electrocardiograma ‘ECG’, electromiografía ‘EMG’, glucómetro, sensor galvánico para la piel, presión arterial y posición del paciente (Hacks, 2013). En la Figura 2.1 se puede apreciar los diferentes sensores que pueden se adaptados a la tarjeta.

Figura 2.1: Tipos de sensores compatibles con e-Health V2.0.
Fuente: (Hacks, 2013).

El e-Health V2.0 es una plataforma de hardware libre, dotada de la instrumentación básica para la adecuación de señales que provienen de los sensores, y, está diseñada con el propósito de ayudar a investigadores, desarrolladores y estudiantes a adquirir señales biomédicas del cuerpo, resultando ser una alternativa económica para la experimentación e implementación de proyectos biomédicos, facilitando la manipulación de dicha información.

La distribución de los sensores en la parte superior y posterior de la plataforma e-Health V2.0 se puede apreciar en la Figura 2.2 y en la Figura 2.3 respectivamente.

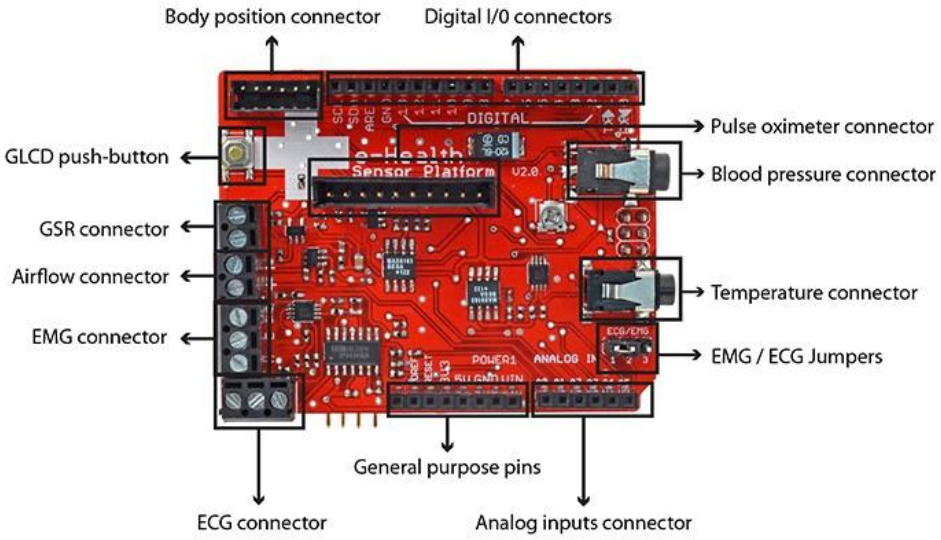


Figura 2.2: Distribución de sensores en plataforma e-Health cara superior.
Fuente: (Hacks, 2013).

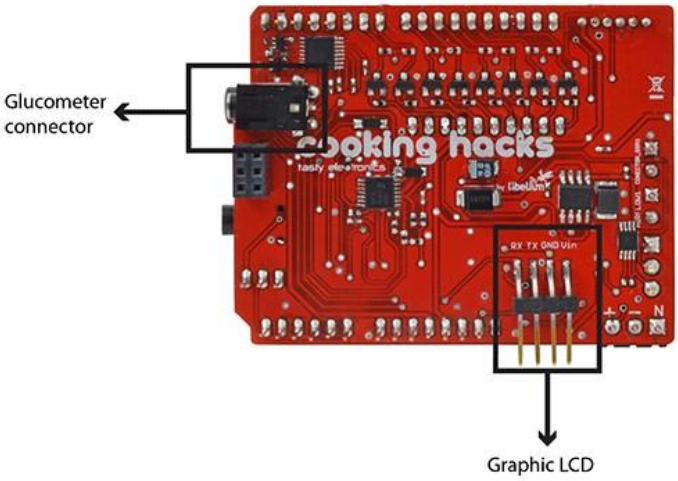


Figura 2.3: Distribución sensores en plataforma e-Health cara posterior.
Fuente: (Hacks, 2013).

2.1.1.1 Funciones generales de la plataforma e-Health

Como se mencionó anteriormente, la plataforma e-Health puede adquirir información de diez parámetros distintos del cuerpo, haciendo uso de sensores que van conectados a la tarjeta, y para ello, también se dispone de librerías que el fabricante pone al alcance de los usuarios desde su página web, estas librerías realizan las conversiones necesarias para obtener una lectura correcta de los sensores, haciendo más sencilla la obtención de los datos. El uso de las funciones dependerá del tipo de proyecto que se vaya a realizar y los parámetros que se desee medir, a continuación en la Tabla 2.1 se lista las funciones[1] disponibles que deben ser añadidos al código de la aplicación.

[1] Para más información sobre el uso de las librerías; dirigirse a la página web de (Hacks, 2013)

Tabla 2.1: Funciones de librería e-Health.

Sensor	Función	Descripción
Pulsioxímetro (SPO2 y RPbpm)	initPulsioximeter()	Inicializa el sensor
	readPulsioximeter()	Lee un valor del sensor
	getBPM()	Devuelve los latidos por minuto del corazón
	getOxygenSaturation()	Devuelve la saturación de oxígeno en%
ECG y EMG	getECG()	Devuelve un valor que representa el potencial del
	getEMG()	Devuelve un valor que representa el potencial del
Flujo de aire	getAirFlow()	Devuelve un valor que representa el flujo de aire
	airFlowWave()	Imprime la forma de onda del flujo de aire en un
Temperatura	getTemperature()	Devuelve el valor de la temperatura corporal
Presión arterial	initBloodPressureSensor()	Inicializa el sensor
	getBloodPressureSensor()	Obtiene la información almacenada en el sensor
	getSystolicPressure(i)	Devuelve el valor de la presión diastólica almacenada en la variable i
	getDiastolicPressure(i)	Devuelve el valor de la presión sistólica
Posición	initPositionSensor()	Inicializa el sensor de posición (acelerómetro)
	getBodyPosition()	Obtiene el valor de la posición del cuerpo
	printPosition()	Imprime la posición actual del cuerpo
Respuesta galvanométrica de la piel	getSkinConductance()	Obtiene el valor de la conductancia de la piel
	getSkinResistance()	Obtiene el valor de la resistencia de la piel
	getSkinConductanceVoltage()	Obtiene el valor de la conductancia de la piel en
Glucómetro	readGlucometer()	Lee los valores almacenados en el
	GetGlucometerLength()	Obtiene el valor almacenado en el
	numberToMonth()	Convierte la variable month a string

Fuente: (Hacks, 2013).

2.1.2 Arduino Uno

Arduino, al igual que la tarjeta e-Health, es una plataforma libre muy utilizada para una variedad de proyectos debido a su versatilidad y bajo costo, además, resulta ser una herramienta sencilla para aquellas personas que recién están incursionando en el mundo de la electrónica y para agilizar el proceso en la realización de proyectos para aquellos más experimentados. Arduino posee su propio entorno de desarrollo 'IDE' (por sus siglas en ingles) el cual corre bajo el lenguaje de C++ y puede ser descargado desde la página web del proveedor. Éste IDE permite configurar los periféricos internos del microprocesador tales como entradas o salidas digitales, conversor analógico/digital 'A/D', moduladores de ancho de pulso 'PWM', comunicación serial, entre otros, según las necesidades del proyecto.

El Arduino Uno R3, concretamente hablando, es una tarjeta basada en el microcontrolador ATMega328P el cual posee 14 pines digitales que pueden ser configurados como entradas o salidas (I/O por sus siglas en Inglés) y de los cuales seis pueden ser utilizados como salidas PWM, además, posee seis entradas analógicas con un conversor A/D de diez bits y trabaja con un cristal de 16MHz. A diferencia con otras plataformas, no posee un chip controlador de USB a serial sino, por el contrario, cuenta con un ATMega16U2 (ATMega8U2 para las versiones anteriores) programado como convertidor de USB a serial. (Arduino, 2015).

El Arduino Uno R3 puede ser alimentado directamente desde el puerto USB, el mismo que se utiliza para su programación, o por una fuente externa según las especificaciones que se muestran en la Tabla 2.2, además, es capaz de comunicarse con otras plataformas mediante el protocolo serial y también cuenta con pines que tienen características especiales para distintos propósitos, la Tabla 2.3 resume las funciones de estos pines.

Tabla 2.2: Especificaciones técnicas del Arduino Uno.

Características	Descripción
Microcontrolador	ATmega328P
Voltaje de operación	5V
Voltaje de entrada recomendado	7 - 12V
Rango límite de voltaje de entrada	6 - 20V
Pines Digitales I/O	14 (6 como salidas PWM)
Salidas digitales PWM	6
Entradas Analógicas	6
Corriente en DC por pin I/O	20 mA
Corriente en DC para el Pin 3.3V	50 mA
Memoria Flash	32 KB (ATmega328P), 0.5KB utilizados por el bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de Reloj	16 MHz
Largo	68.6 mm
Ancho	53.4 mm
Peso	25 g

Fuente: (Arduino, 2015).

Tabla 2.3: Características especiales de los pines.

Pines	Características
0(RX), 1(TX)	Estos pines sirven para la recepción (RX) y transmisión (TX) de información TTL por el puerto serial, cuando estos están siendo utilizados no pueden ser configurados como I/O.
2, 3	Pueden ser configurados como interrupciones para detectar eventos externos que requieran una atención prioritaria.
3, 5, 6, 9, 10, 11	Proveen un salida PWM de 8 bits
10(SS), 11(MOSI), 12 (MISO), 13(SCK)	Esto pines soportan la comunicación SPI (Serial Peripherical Interface), la cual es una comunicación síncrona maestro a esclavo.
A0(SDA), A5(SCL)	Sirven para la comunicación TWI o comúnmente conocida como I^2C.

Fuente: (Arduino, 2015).

2.1.3 Raspberry Pi 2 model B

Con el paso del tiempo, el avance tecnológico ha permitido reducir el tamaño de los componentes electrónicos a tal punto que ha sido posible fabricar miniordenadores, y, aunque todavía no poseen las mismas características de los ordenadores convencionales, disponen de un buen desempeño y rendimiento para ejecutar tareas como navegar por Internet, reproducción de video, juegos sencillos, centro multimedia, programación, entre otras.

Figura 2.4: Raspberry Pi 2 model B.
Fuente: (Upton, 2015).

Es así como la fundación Raspberry Pi, procedente del Reino unido (UK), desarrolló el miniordenador que lleva el mismo nombre, con el objetivo de crear un dispositivo de bajo costo que permita promover la enseñanza, en jóvenes y adultos, principalmente en el área de las computadoras lo cual permite a personas de todas las edades explorar el mundo computación, y aprender lenguajes de programación tales como Scratch y Python, además, puede hacer todo lo que se esperaría de un ordenador de escritorio pero con la ventaja de poder interactuar con el mundo debido a sus pines de propósito general, en los cuales pueden conectar sensores para obtener sus lecturas y trabajar con dichos datos. (Raspberry, 2015).

La Raspberry Pi ha sufrido varias actualizaciones desde que se lanzó su primera versión hasta llegar a su versión actual, la Raspberry pi 2 modelo B, la misma que posee un procesador más potente y el doble de memoria RAM respecto a su predecesor, para más detalle referirse a la Tabla 2.4. Para interactuar con el

dispositivo solo es necesario conectar un teclado y mouse USB y un monitor a través del puerto HDMI. (Raspberry, 2015).

Como cualquier ordenador, se necesita de un sistema operativo para funcionar, por lo que el fabricante recomienda Raspbian , que es una distribución de Linux basada en Debian[2], la misma que está optimizada para el hardware del dispositivo y que incluye más de 35000 paquetes y programas pre compilados en un formato fácil de instalar. Aunque, Raspbian es el sistema operativo oficial, también existen otras distribuciones de Linux, entre ellas Ubuntu, así como también Windows 10 e incluso Android, aunque este último aún no se recomienda debido a su inestabilidad. (Raspberry, 2015).

Tabla 2.4: Especificaciones Raspberry pi 2.

Característica	Descripción
SoC	Broadcom BCM2836
CPU	ARM11 ARMv7 ARM Cortex-A7 4 núcleos @ 900 MHz
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
RAM	1 GB LPDDR2 SDRAM 450 MHz
Salida de video	HDMI 1.4 a 1920x1200 píxeles
Ethernet	10/100Mbps
Consumo	5v, 900mA
Dimensiones	85,60x56,5 mm
Peso	45gr

Fuente: (Upton, 2015).

2.2. Lectura de Sensores

Para la adquisición de las señales fisiológicas del cuerpo tales como el ECG, SPO2, frecuencia cardíaca y temperatura corporal se emplea el uso de la tarjeta e-Health debido que, como se mencionó anteriormente, posee la instrumentación necesaria para ese fin y que se describirá más adelante, la tarjeta va montada sobre el Arduino tal como se muestra en la Figura 2.5, por otra parte, para obtener el peso y para el control del sistema de calefacción se requiere de hardware adicional que se describe a continuación.

[2] Debian es un sistema operativo libre a base de Linux creado por una comunidad de programadores.

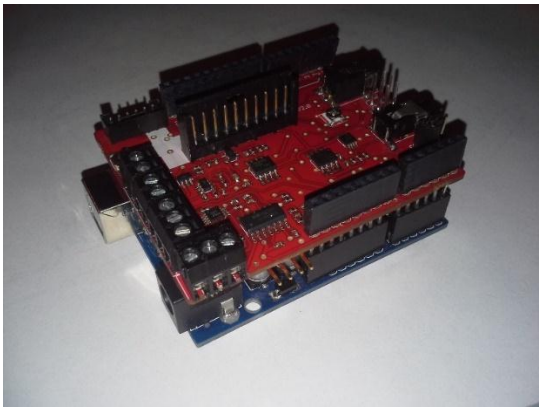


Figura 2.5: Montaje de la tarjeta e-Health.

2.2.1 Balanza

Para obtener la medición de peso del paciente se emplean celdas de carga, las cuales producen una diferencia de potencial entre dos de sus terminales debido a la interacción con una fuerza externa aplicada. Las celdas de carga son sensores resistivos que están constituidas internamente por galgas extensiométricas, cuya característica es variar su resistividad en función de la deformación longitudinal que sufre la misma, generalmente, las celdas de carga poseen cuatro galgas dispuestas en puente y orientadas en sentido opuesto respecto a la galga perteneciente a la misma rama para, de esta manera, mejorar su linealidad y compensar la variación de resistencia producida por los cambios de temperatura, ver Figura 2.6, esta configuración se denomina puente de Wheatstone y permite medir las pequeñas variaciones de voltaje que se produce entre los puntos S+ y S-.

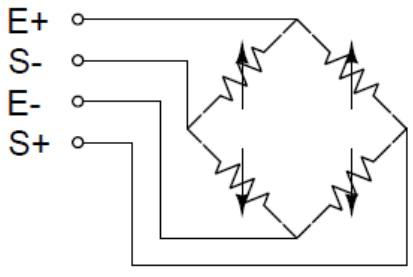


Figura 2.6: Configuración interna de una celda de carga.

2.2.1.1 Celdas de carga

Para el circuito de pesaje se emplean las celdas de carga de propósito general SEN-10245, éstas celdas soportan un peso de 50Kg y son comúnmente utilizadas en

basculas digitales de uso doméstico, además, poseen la particularidad de estar conformadas por dos galgas configuradas en partidor de tensión (medio puente), como muestra la Figura 2.7, por lo que es necesario de dos resistencias externas de igual denominación o de otra celda de carga de características similares para mantener el puente en equilibrio, es decir; que la diferencia de potencial entre S+ y S- sea igual a cero. En la Tabla 2.5 se describe las especificaciones técnicas de la galga utilizada.

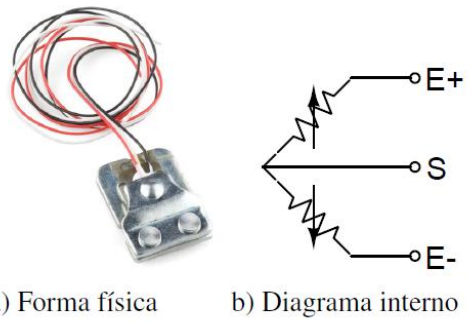


Figura 2.7: Celda de carga SEN-10245.
Fuente: (Olimex & MCI, 2015).

Tabla 2.5: Características de celda de carga SEN-10245.

Característica	Unidad	Valor
Comprehensive Error	mV/V	0.05
Output Sensitivity	mV/V	1.0±0.1
Nonlinearity	%FS	0.03
Repeatability	%FS	0.03
Hysteresis	%FS	0.03
Creep	(3min)%FS	0.03
Zero Drift	(1min)%FS	0.03
Temp. Effect on Zero	%FS/10°C	1
Temp. Effect on Output	%FS/10°C	0.05
Zero Output	mV/V	±0.1
Input Resistance	K	1000±20
Output Resistance	K	1000±20
Insulation Resistance	MK	≥5000
Excitation Voltage	V	≤10
Operation Temp. Range	°C	0–+50
Overload Capacity	%FS	150

Fuente: (Olimex & MCI, 2015).

Para facilitar el proceso, se emplea una báscula digital de baño la cual contiene cuatro celdas de carga de medio puente, debido que el circuito electrónico utilizado por el fabricante no es de interés se lo descarta y solo se utiliza la estructura y los sensores, además, se conserva el conexionado empleado por el fabricante para completar el puente de Wheatstone tal como se ilustra en la Figura 2.8.

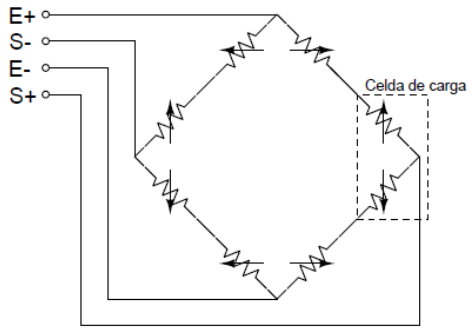


Figura 2.8: Conexionado de celdas de carga.

2.2.1.2 Ajuste del puente

Como se mencionó, las celdas de carga utilizadas deben ser de la misma denominación para conservar el equilibrio del puente, sin embargo, en la práctica esto no es así, ya que, debido a la tolerancia que presentan los materiales con los que se fabrican se genera una pequeña variación, en el orden de los mili voltios, por lo que es necesario de un circuito adicional para mitigar esta variación. La Figura 2.9 muestra el circuito necesario para el ajuste del puente, los valores de las resistencias puede ser de cualquier denominación pero se recomienda que cumpla con las siguientes condiciones: $\frac{E-Vg}{Rb}R4 > |Vso|$ y $\frac{Vg}{Rb}R1 > |Vso|$, el valor de Ra puede ser del mismo valor o un valor aproxima siempre que no represente una carga excesiva para la fuente. (Sánchez, 2007).

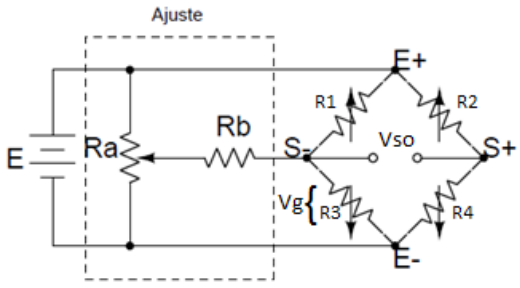


Figura 2.9: Circuito de ajuste para puente de Wheatstone.

2.2.1.3 Acondicionamiento de la señal

Adicionalmente, para poder realizar una lectura más acertada es necesario realizar un acondicionamiento de la señal mediante amplificadores operacionales, para lo cual se utiliza la configuración en seguidor de tensión (o buffer), como se puede observar en la Figura 2.10, con el propósito de acoplar impedancias debido a la magnitud muy pequeña de la señal a adquirir, posterior a este paso, es necesario amplificar la señal a valores más mensurables por medio de amplificadores de instrumentación, los cuales presentan un alto rechazo al ruido, por lo que se emplea el amplificador operacional INA131AP cuya ganancia mínima es de 100 , se debe tener presente que mientras más alta sea la ganancia también se amplifica el ruido. La ganancia utilizada es de 300 y se calcula mediante la ecuación 2.1[3].

$$G = 100 + \frac{250K\Omega}{RG} \qquad (2.1)$$

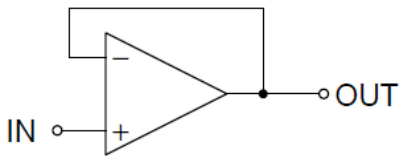


Figura 2.10: Configuración buffer.

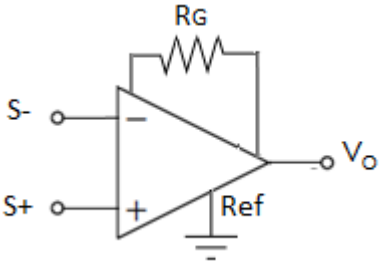


Figura 2.11: Amplificación con OPAM de instrumentación INA131AP.

2.2.1.4 Filtrado

Para eliminar posibles interferencias que puedan contaminar la señal, se aplican dos filtros pasa bajos de segundo orden en serie y con una frecuencia de corte de 4Hz

[3] Extraído de la hoja de datos, disponible en:
http://pdf.datasheetcatalog.com/datasheet2/6/0epd04owif3dzwsepjkzqgsw98yy.pdf

cada uno, que se calcula con la ecuación 2.2, para posteriormente ingresar la señal al microprocesador del Arduino. Los filtros están configurados para una ganancia de 1.79 y 1.51 respectivamente y se calcula con la ecuación 2.3. La Figura 2.12 muestra el esquema utilizado.

$$f_c = \frac{1}{2\pi RC} \qquad (2.2) \qquad \mathrm{G} = 1 + \frac{RF}{R1} \qquad (2.3)$$

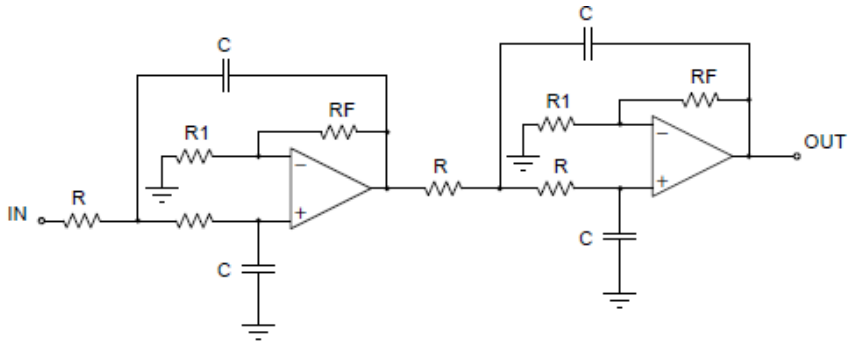


Figura 2.12: Filtro pasa bajo Butterworth de cuarto orden.

2.2.1.5 Fuente de alimentación

Debido que, la mayoría de amplificadores operaciones y amplificadores de instrumentación necesitan de una fuente de alimentación simétrica para su funcionamiento, se utiliza el circuito integrado TC7660 de Microchip configurado como inversor de voltaje, empleando el esquema de la Figura 2.13, evitando de esta forma tener que implementar un transformador.

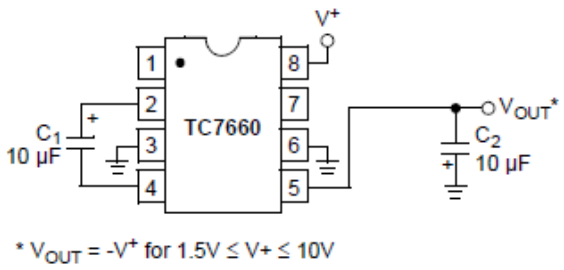


Figura 2.13: Circuito inversor para fuente simétrica.

Fuente: (Microchip, 2011).

2.2.2 Control de temperatura

Al igual que el circuito de balanza, se emplea un puente de Wheatstone para medir la diferencia de potencial entre los puntos TEMP+ y TEMP- producida por el cambio de resistencia del sensor debido a la temperatura, el puente utiliza el circuito integrado MAX6163 para regular el voltaje de referencia a tres voltios, de igual manera, al ser una variación muy pequeña es necesario acondicionar la señal mediante un amplificador de instrumentación, INA321EA, que puede trabajar con una fuente de tensión simple en un rango de tres a cinco voltios, la Figura 2.14 muestra el esquema utilizado.

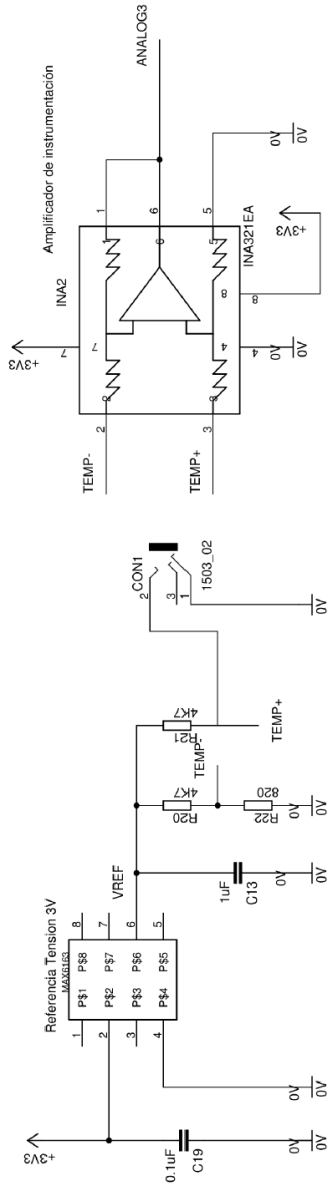


Figura 2.14: Circuito de temperatura.

Fuente: (Hacks, 2013).

2.2.2.1 Circuito de potencia

Para mantener la temperatura del paciente en un rango permisible se emplea una resistencia de calor de 600w la cual varia su potencia a través del TRIAC BT136-600E, mediante el esquema de la Figura 2.15, el mismo que soporta hasta 10A con la ayuda de un disipador apropiado que evita el sobrecalentamiento del integrado, la potencia entregada varía en función de la temperatura del paciente por medio de un control PID, el mismo que genera una señal PWM que controla a un opto-acoplador, el moc3041, que aísla la etapa de control. La señal de control trabaja a una frecuencia de 30Hz lo que garantiza que pase todo el ciclo de la onda de la red eléctrica cuando se detecte el cruce por cero.

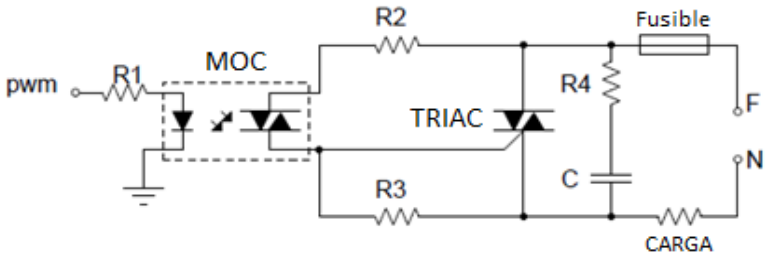


Figura 2.15: Control de potencia para calefactor.

2.2.3 Circuito ECG

Para capturar el potencial del corazón se realiza una medición *in vitro*[4] a través de electrodos de cloruro de plata (AgCl) , los mismos que entran en contacto con la piel del paciente mediante un electrólito que permite el intercambio de iones, para la distribución de los electrodos se aplica el diagrama de Einthoven (Gordillo Padilla, 2014), el potencial recogido se transmite por medio de cables apantallados que reducen la contaminación de la señal hasta llegar al circuito de acondicionamiento de la tarjeta e-Health que se ilustra en la Figura 2.16.

.

[4] Hace referencia a la medición del potencial del cuerpo con sensores no invasivos que se adhieren a la superficie de la piel.

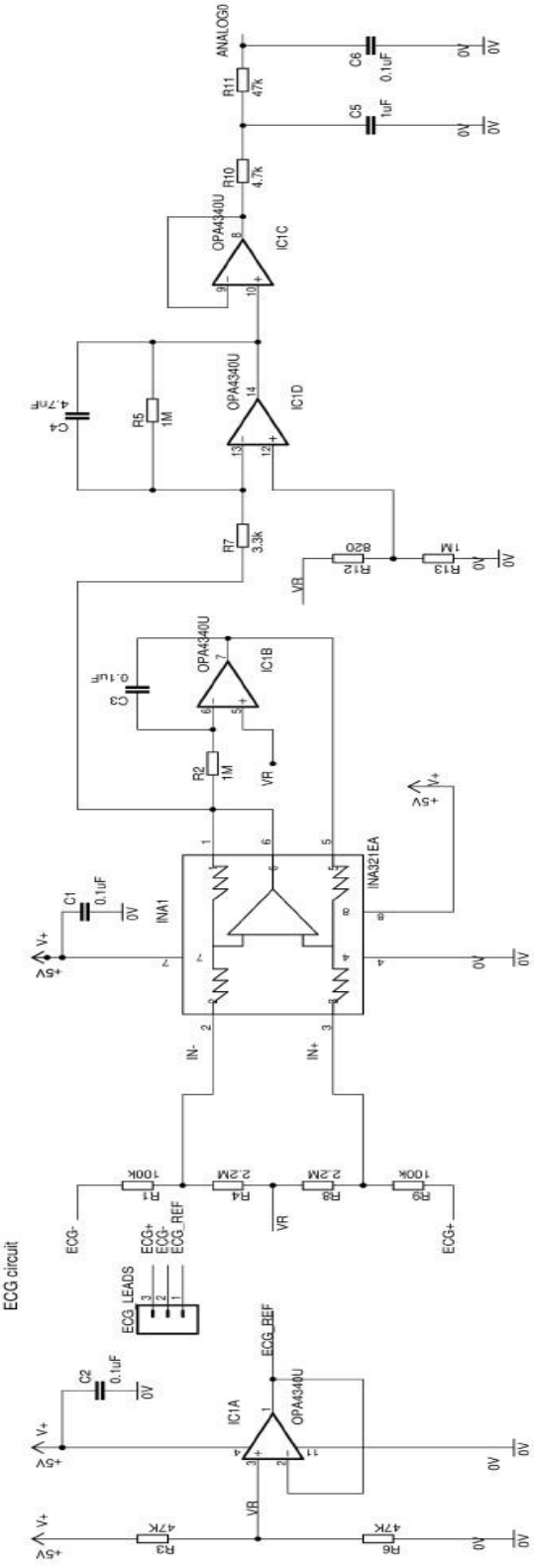


Figura 2.16: Circuito de acondicionamiento de ECG.

Fuente: (Hacks, 2013).

2.2.3.1 Filtro ECG

Debido que la instrumentación de la tarjeta e-Health es básica y que el potencial del corazón es muy susceptible al ruido, es necesario limpiar la señal mediante filtros, para tal efecto se emplea un filtro Notch en serie con un filtro pasa bajos de segundo orden configurados para una frecuencia de corte de 60Hz y 22Hz respectivamente, empleando la ecuación 2.2, el esquema utilizado para el filtro pasa bajos es similar al de la Figura 2.12 pero calculado para una ganancia de 1.67 debido a la atenuación que sufre la señal al pasar por el filtro. El circuito completo empleado para el filtrar la señal del ECG se muestra en la Figura 2.17.

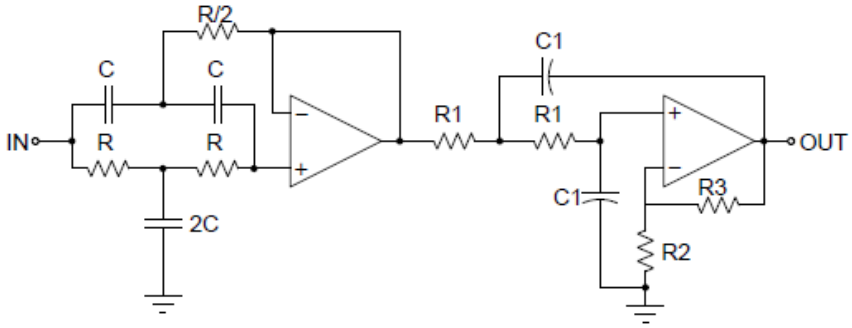


Figura 2.17: Filtro para ECG.

2.2.4 Sensor pulsioxímetro

El sensor permite obtener la frecuencia cardíaca y medir, de forma no invasiva, la concentración de oxígeno presente en la hemoglobina sanguínea mediante sensores luminosos que emiten dos tipos de luz roja a diferentes longitudes de onda y la compara con un valor de referencia llamado saturación parcial de oxígeno (SpO2). (Gordillo Padilla, 2014).

El sensor muestra la información obtenida en un display de segmentos, del cual se obtienen las señales que saturan los transistores del circuito que se muestra en la Figura 2.18 y cuyas combinaciones son interpretadas en el Arduino para decodificar la información mediante las funciones correspondientes de la Tabla 2.1.

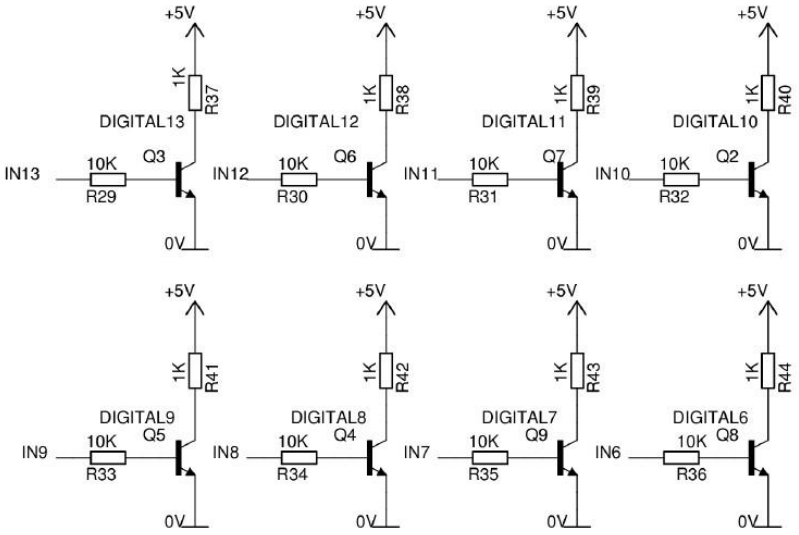


Figura 2.18: Circuito de captura para pulsioxímetro

Fuente: (Hacks, 2013).

2.3. Procesamiento de datos

Para la creación del sistema se hace uso de las características y capacidades de los distintos equipos que se han descrito antes, los mismos que están distribuidos como presenta el diagrama de bloques de la Figura 2.19.

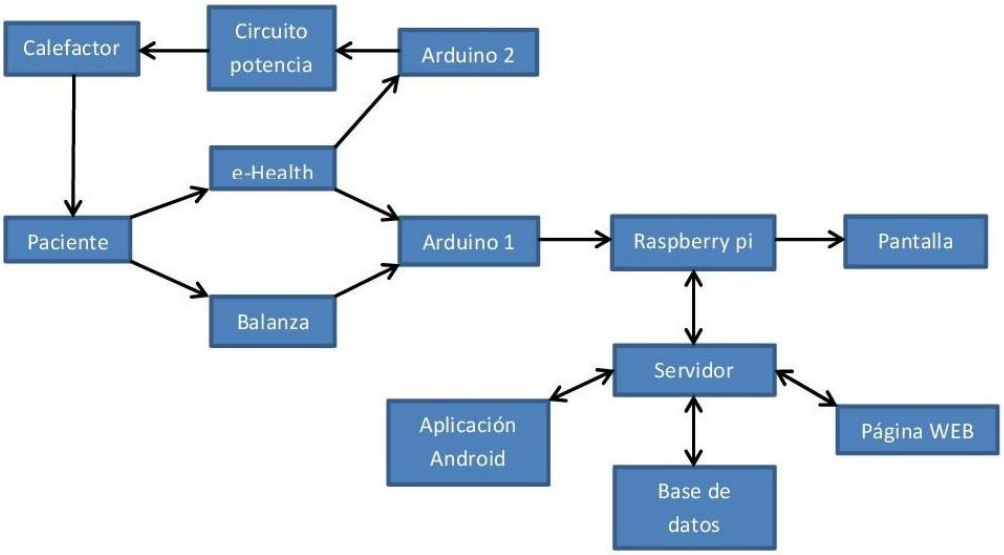


Figura 2.19: Diagrama del sistema.

2.3.1 Procesamiento Arduino

Como se pudo observar en la Figura 2.19, el sistema dispone de dos unidades Arduino numeradas del uno al dos respectivamente, cada unidad realiza una tarea específica e independiente respecto a la otra.

2.3.1.1 Arduino 1

En el Arduino1 va montada la plataforma e-Health y es la interfaz encargada de procesar las señales que provienen de la misma, así como también, la información que proviene del circuito de balanza. Las señales de ECG, temperatura y peso ingresan al microprocesador a través de los puertos analógicos A1, A3 y A5 respectivamente, las lecturas son procesadas por el conversor A/D y divididas entre 4, cada una, para que puedan ser convertidas a un Byte, los datos de la frecuencia cardíaca y saturación de oxígeno provenientes del sensor, pulsioxímetro, son obtenidas a través de las funciones expuestas en la Tabla 2.1.

Los datos obtenidos son almacenados en variables, de tipo entero, que han sido declaradas previamente y se transmiten una tras otra, formando una trama, hacia el procesador central (Raspberry Pi) mediante comunicación serial, siempre que este último realice una petición a través de la letra ‘t’, para de esta forma sincronizar la comunicación entre ambos dispositivos. La Figura 2.20 presenta el diagrama de flujo empleado para el procesamiento de datos.

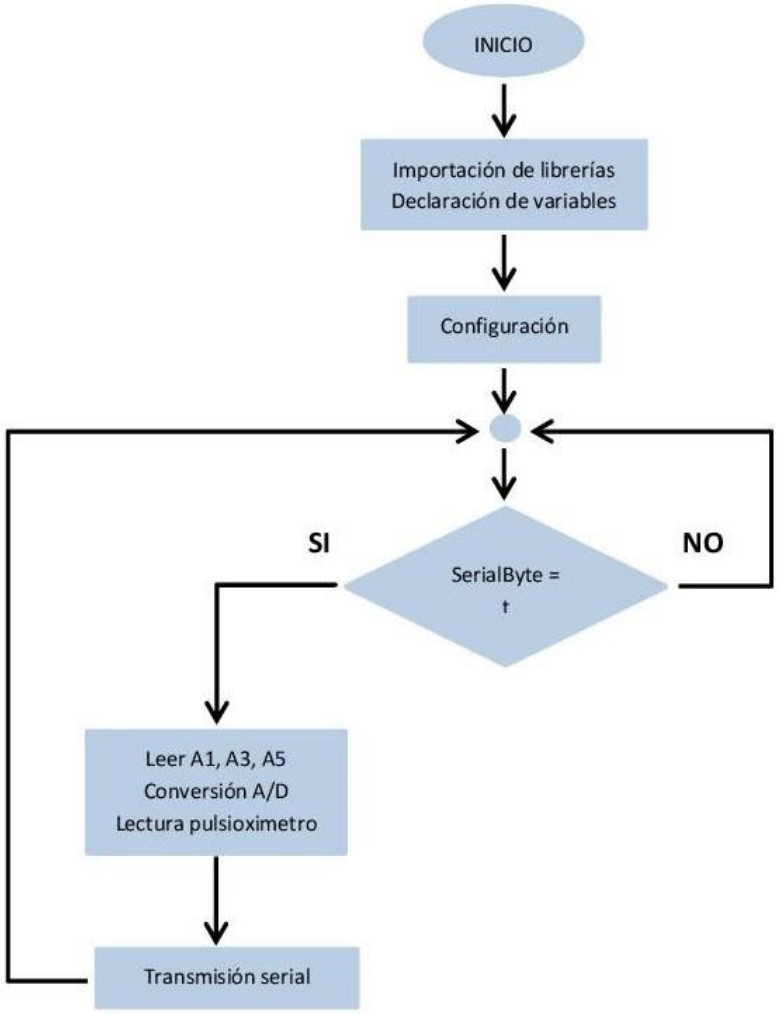


Figura 2.20: Diagrama de flujo Arduino1.

2.3.1.2 Arduino 2

El Arduino2 es la interfaz que permite controlar la potencia entregada al calefactor, y está programado empleando el diagrama de flujo de la Figura 2.21, el IDE de Arduino facilita la implementación de un control PID mediante una librería que cumple ese fin, para ver el código utilizado refiérase al Anexo 4.

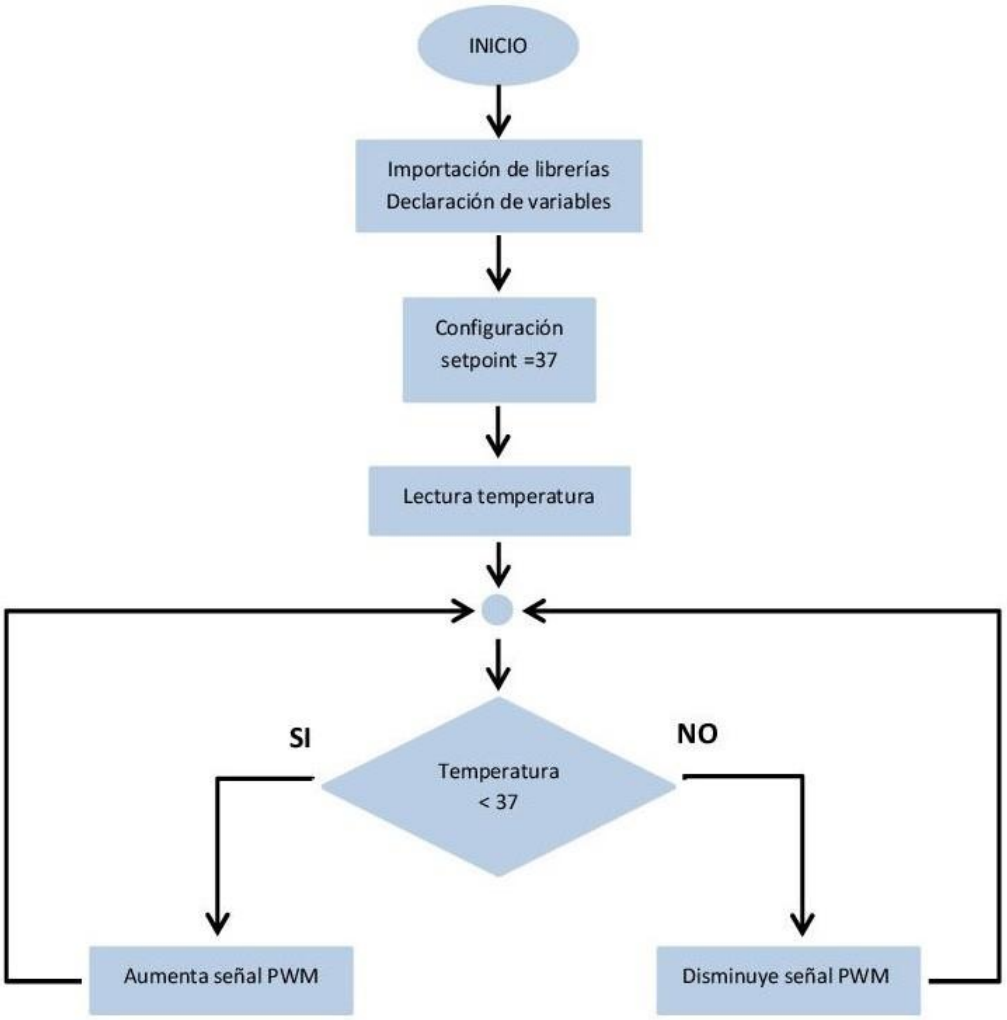


Figura 2.21: Diagrama de flujo Arduino 2.

2.3.2 Procesamiento en Raspberry

Aprovechando las ventajas que ofrece la Raspberry Pi, se procede a realizar un programa que recibe la información enviada desde el Arduino1 y que sea capaz de visualizarla y transmitirla hacia el servidor. El programa está desarrollado en Python debido que es un lenguaje de programación interpretado que permite trabajar de forma rápida y sin perder eficiencia, está constituido por tres pantallas que se describen más adelante y presenta el funcionamiento descrito en el diagrama de bloques de la Figura 2.22. Cabe mencionar que para la creación de la interfaz gráfica se emplea la librería TKinter y para la comunicación que efectúa cada pantalla con el servidor se realiza a través de la librería XmlRpc.

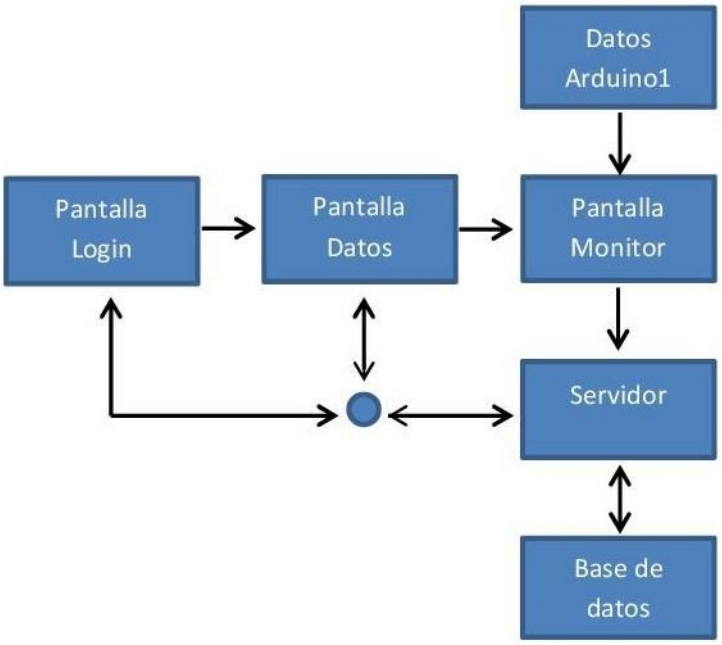


Figura 2.22: Sistema de procesamiento Raspberry.

2.3.2.1 Pantalla login

La pantalla login tiene la función de autenticar los datos del personal a cargo del paciente, para lo cual, consta de dos campos de texto, usuario y contraseña, donde se ingresan los datos correspondientes que se deben crear previamente en la base de datos. Al hacer clic en el botón INGRESAR se envía la información hacia el servidor, el mismo que se comunica con la base de datos para buscar al usuario correspondiente, en caso de haber algún dato incorrecto o los campos queden vacíos se genera un mensaje de error, como muestra la Figura 2.23, que alerta al usuario para reingresar los datos, si se presiona el botón CERRAR se cancela el proceso y se cierra la aplicación.

Figura 2.23: Pantalla login.

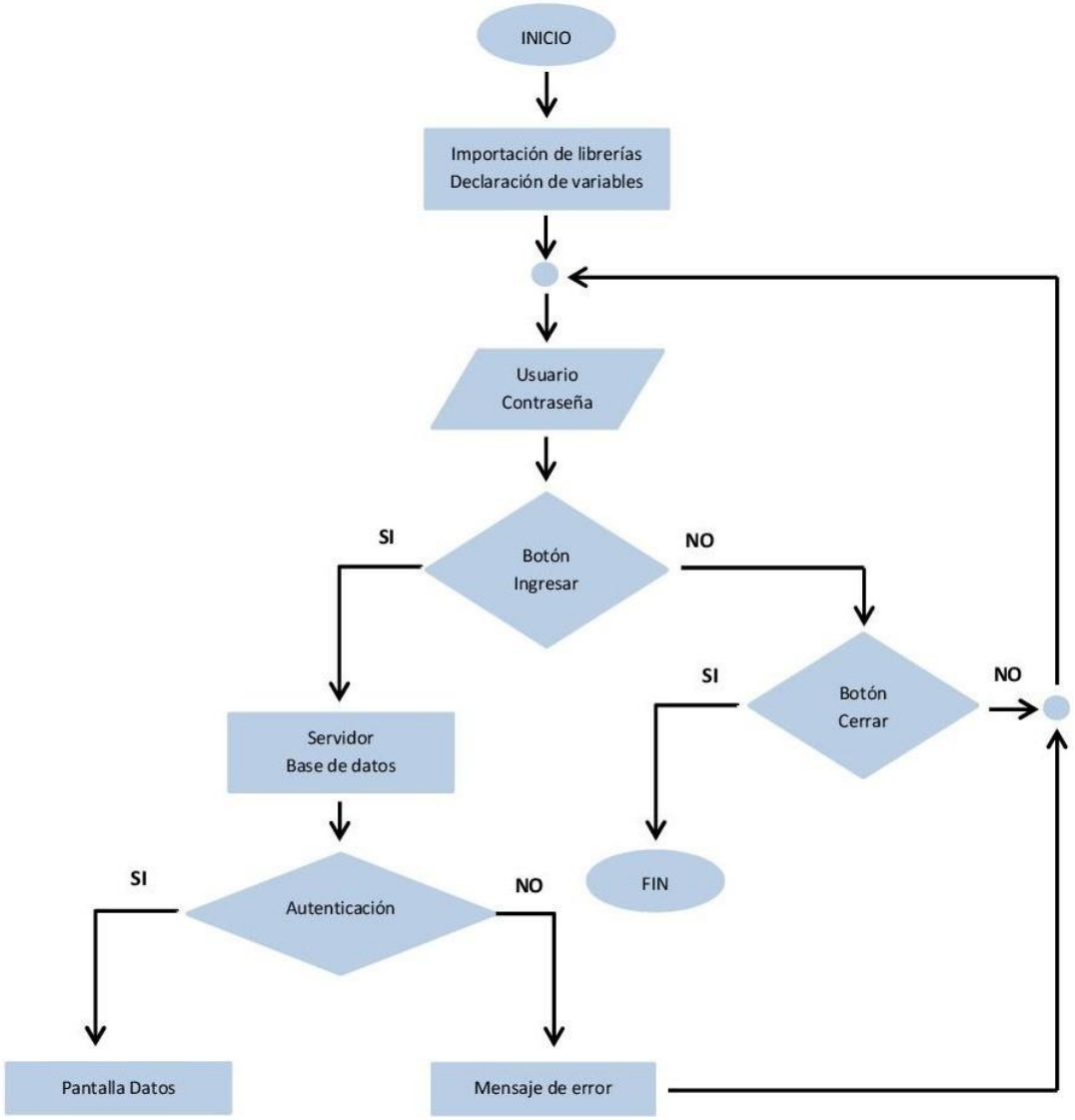


Figura 2.24: Diagrama de flujo para pantalla login.

2.3.2.2 Pantalla datos

Luego de la autenticar al usuario, se despliega la pantalla datos cuya función es crear el registro con la información del paciente, la pantalla está formada por los campos nombre, apellido, fecha de nacimiento y doctor encargado que deben ser llenados para poder pasar a la siguiente pantalla caso contrario se mostrara un mensaje de aviso. Al presionar el botón INGRESAR los datos son enviados hacia el servidor, donde nuevamente se comunica con la base de datos para almacenar la información en los campos de la tabla correspondiente y se le asigna un código único que identificará al paciente.

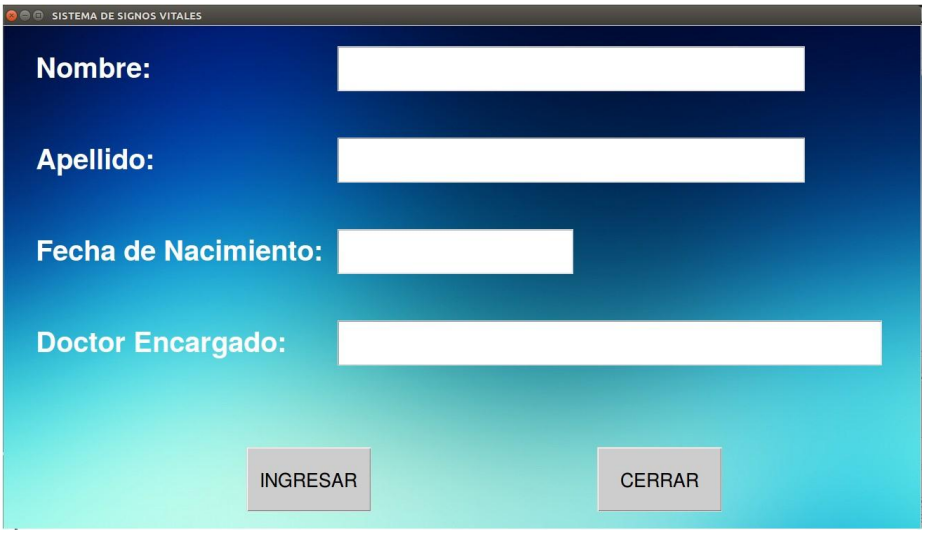


Figura 2.25: Pantalla datos.

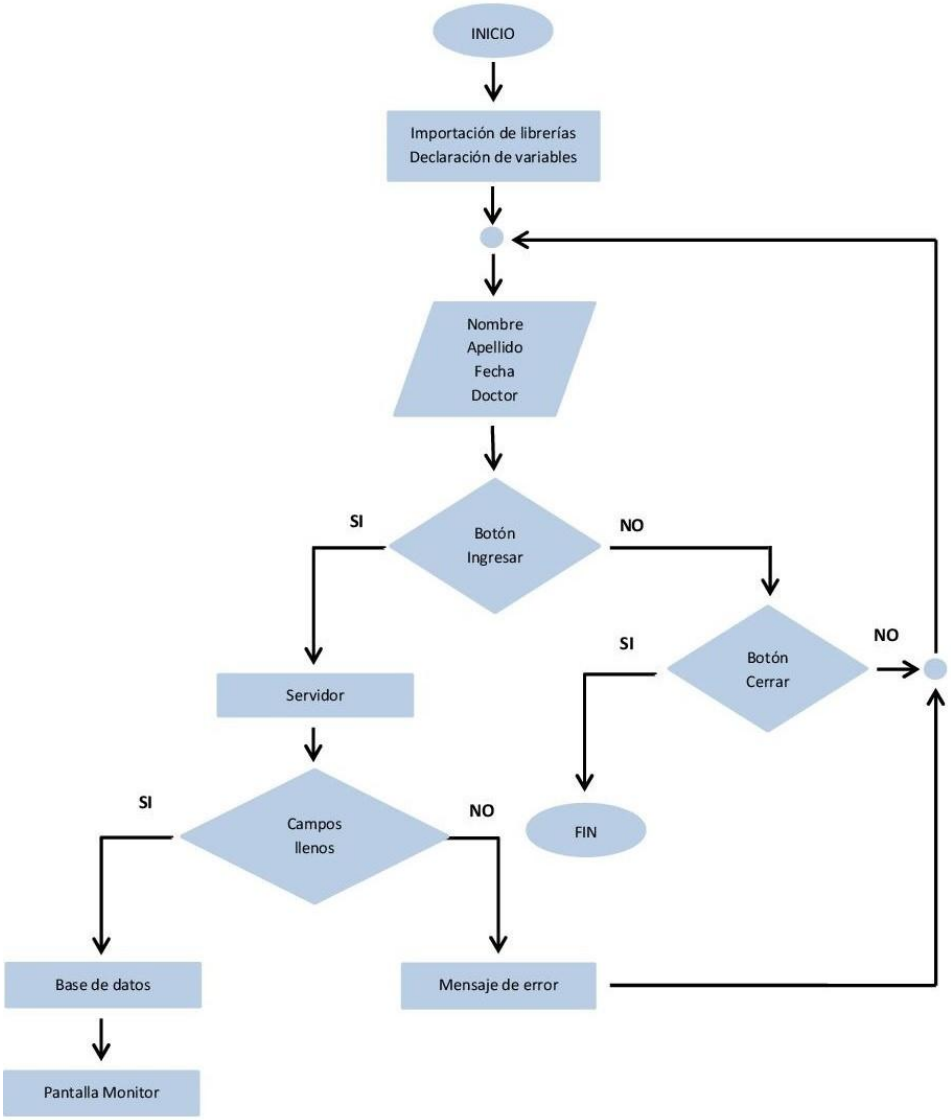


Figura 2.26: Diagrama pantalla datos.

2.3.2.3 Pantalla monitor

Una vez creado el registro, se invoca a la pantalla monitor la cual muestra los signos vitales del paciente que provienen de los sensores, también, se recupera la información y el código generado en el paso anterior. La pantalla dispone de cuatro botones que son: INICIAR, CERO, PARAR y SALIR como se aprecia en la Figura 2.27. Al presionar el botón INICIAR el proceso de adquisición de datos comienza, el programa recibe la trama enviada desde el Arduino1 y la separa para realizar las conversiones y cálculos correspondientes para interpretar la información y los almacena temporalmente en un buffer, cuando el buffer se llena se procede al envío de datos hacia el servidor, también se verifica si no se ha generado alguna alerta y muestra la información en la pantalla, para el grafico del ECG se utiliza la librería *matplotlib.* Todo esto se realiza de forma simultánea debido que cada proceso se realiza en un hilo diferente para que no interfieran los unos con los otros.

Para calibrar la balanza a cero se presiona el botón con la misma etiqueta, esto se debe realizar antes de poner a reposar al paciente, si se desea detener la lectura de datos sin salir del sistema se presiona el botón PARAR y se puede retomar el proceso volviendo a presionar el botón INICIAR. Cuando el paciente es dado de alta se finaliza el proceso mediante el botón SALIR, así el sistema queda listo para ser utilizado con otro paciente, el diagrama de la Figura 2.28 ilustra el funcionamiento de la pantalla monitor.

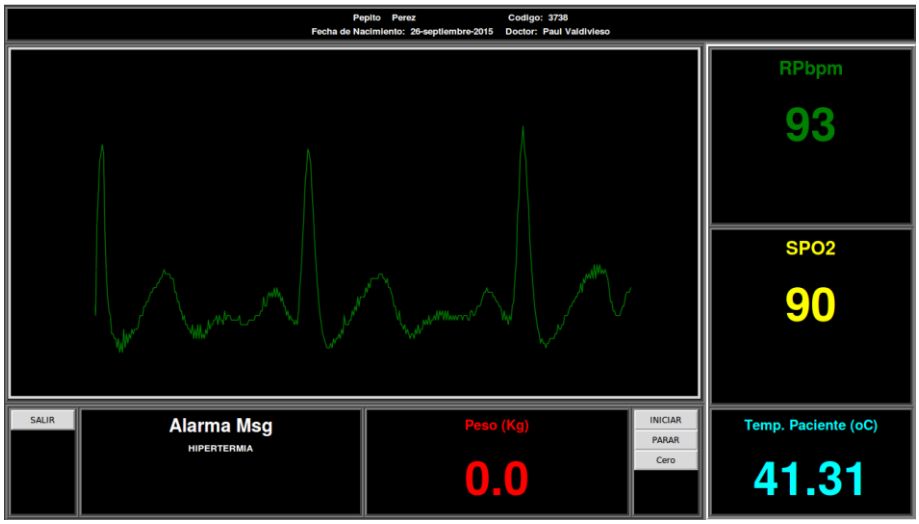


Figura 2.27: Pantalla monitor.

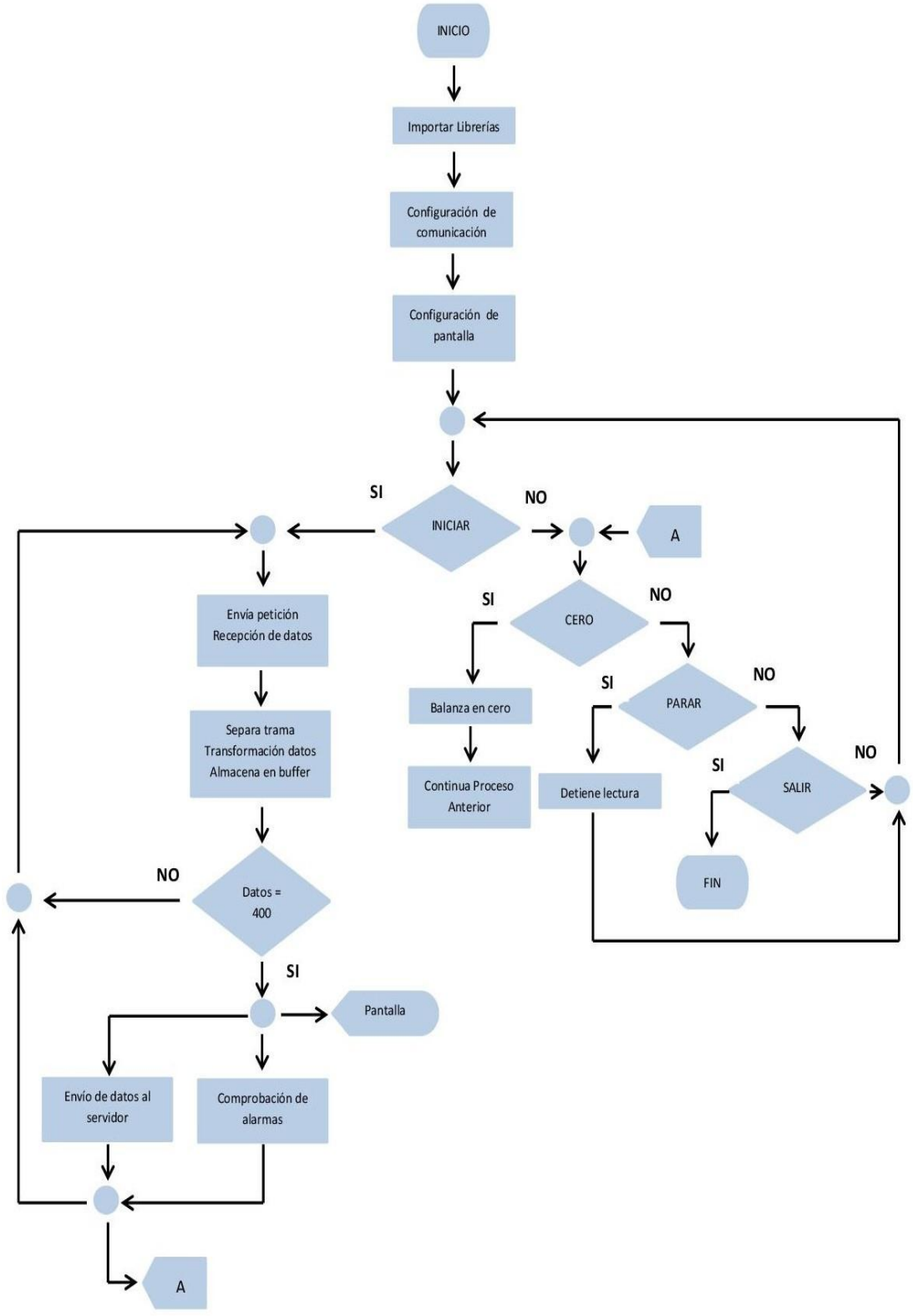


Figura 2.28: Diagrama de flujo pantalla monitor.

CAPÍTULO 3

SERVIDOR WEB Y CLIENTE

Resumen: En éste capítulo se describen los elementos que conforman el programa servidor así como también los programas cliente, dando una breve explicación de su funcionamiento y el propósito que cumplen para dar cohesión al proyecto.

3.1. Servidor WEB

3.1.1 Lenguaje de Programación

Java es un lenguaje de programación orientado a objetos y, como cualquier otro lenguaje, éste tiene sus propias reglas de sintaxis y estructuras, está basado en el lenguaje de programación C con el que comparte ciertas similitudes como los bloques de módulos que se delimitan por llaves y como la declaración, previa al uso, de las variables. La programación en Java se caracteriza por ser paquetes, los cuales contienen a las clases y estos a su vez contienen los métodos, variables, constantes, etc. (Oracle, JavaServer Faces Technology Overview, 2015).

A diferencia de otros lenguajes de programación, cuyas instrucciones son compiladas para que se ejecuten directamente en la CPU, Java crea primero un archivo con extensión (.java) que al ser compilado se trasforma en un código de bytes que se almacena en un archivo de extensión (.class) el cual se ejecuta en la máquina virtual de java 'JVM'. (Oracle, JavaServer Faces Technology Overview, 2015).

3.1.2 Protocolo de Comunicación XML-RPC

Para establecer la conexión entre el servidor y el procesador central se utiliza el protocolo de comunicación Xml-RPC (eXtensible Markup Language - Remote Procedure Calls), el cual es un protocolo de llamada a procedimiento remoto que permite establecer comunicación entre lenguajes de programación impares, es decir; lenguajes distintos, como por ejemplo: Python y Java, también permite hacer

llamadas de procedimientos entre distintas plataformas a través de Internet. (Scripting & Userland, 2011).

El protocolo utiliza HTTP (Hypertext Transfer Protocol) para el transporte de la información por World Wide Web y XML para codificar los datos, el mismo que fue diseñado para ser lo más simple de interpretar pero que permita realizar estructuras de datos de envío lo más complejas posibles. (Scripting & Userland, 2011).

3.1.3 NetBeans

NetBeans es uno de los compiladores de código abierto más utilizados a nivel mundial, fue fundado en el año 2000 por Sun MicroSystems y en los últimos años se ha venido desarrollando dos tipos de productos: el NetBeans IDE y NetBeans Platform. El NetBeans Platform, es una base regular y extensible utilizada por empresas especializadas en el desarrollo de software, para crear pequeñas y grandes aplicaciones de escritorio que se integran fácilmente a cualquier plataforma o sistema operativo que tenga instalada la máquina virtual de Java. (Oracle, Bienvenido a NetBeans y www.netbeans.org, 2015).

Por otra parte, NetBeans IDE es una herramienta para que los programadores puedan escribir, compilar, depurar, y ejecutar programas, aunque está desarrollado en Java también funciona para cualquier tipo de lenguaje de programación de software, además, proporcionan extensiones adicionales que se integran muy fácilmente en la plataforma. El IDE permite crear aplicaciones de Escritorio, Móviles y aplicaciones web así como aplicaciones HTML5 con HTML, JavaScript y CSS e incluye muchas herramientas para desarrollo de PHP y C/C++. (Oracle, Bienvenido a NetBeans y www.netbeans.org, 2015) (Oracle, NetBeans IDE Features, 2015).

3.1.4 Diseño Servidor

El servidor está programado para que establezca la conexión con las distintas etapas y las vincule con la base de datos, que está en el mismo servidor, y con los clientes en Python y Android. La comunicación se realiza mediante el direccionamiento IP a través de un puerto, por donde siempre está a la escucha de alguna petición.

En la Figura 3.1 se muestra el proceso del programa servidor, el mismo que está constituido por la clase principal que lleva el mismo nombre y cinco clases complementarias (Usuario, DatoPaciente, Paciente, DatosSignos y Alta), la clase servidor cumple la función de vínculo entre la petición del cliente y la clase correspondiente. La clase Usuario obtiene los datos ingresados en la pantalla login para el acceso al sistema por parte del personal, el siguiente paso se realiza en la clase DatoPaciente donde la información de registro del paciente es ingresada a la base de datos, lo valida como activo, y los devuelve para ser visualizados en la pantalla monitor y en la clase DatosSignos se registran los valores adquiridos por los sensores, éste proceso se repite continuamente hasta que el paciente es dado de alta mediante la invocación de la clase Alta donde el paciente es puesto como inactivo impidiendo la visualización de los datos en los clientes.

Todo el proceso de consulta hacia la base de datos, que efectúa cada clase, se lo realiza a través de la funciones que ofrece la librería *mysql-connector-java-3.0.17-ga-bin.jar,* la misma que permite extraer la información de los datos requeridos mediante el comando *select nombreTabla from parametro* y cuando se necesita insertar la información a la base de datos se ocupa el comando *insert into nombreTabla values (dato1, dato2,.....datoN)*.

Además, para el intercambio de la información se lo hace mediante el protocolo Xml-Rpc a través de los métodos que ofrecen las librerías org.apche.xmlrpc.WebServer y org.apche.xmlrpc.XmlRpc.

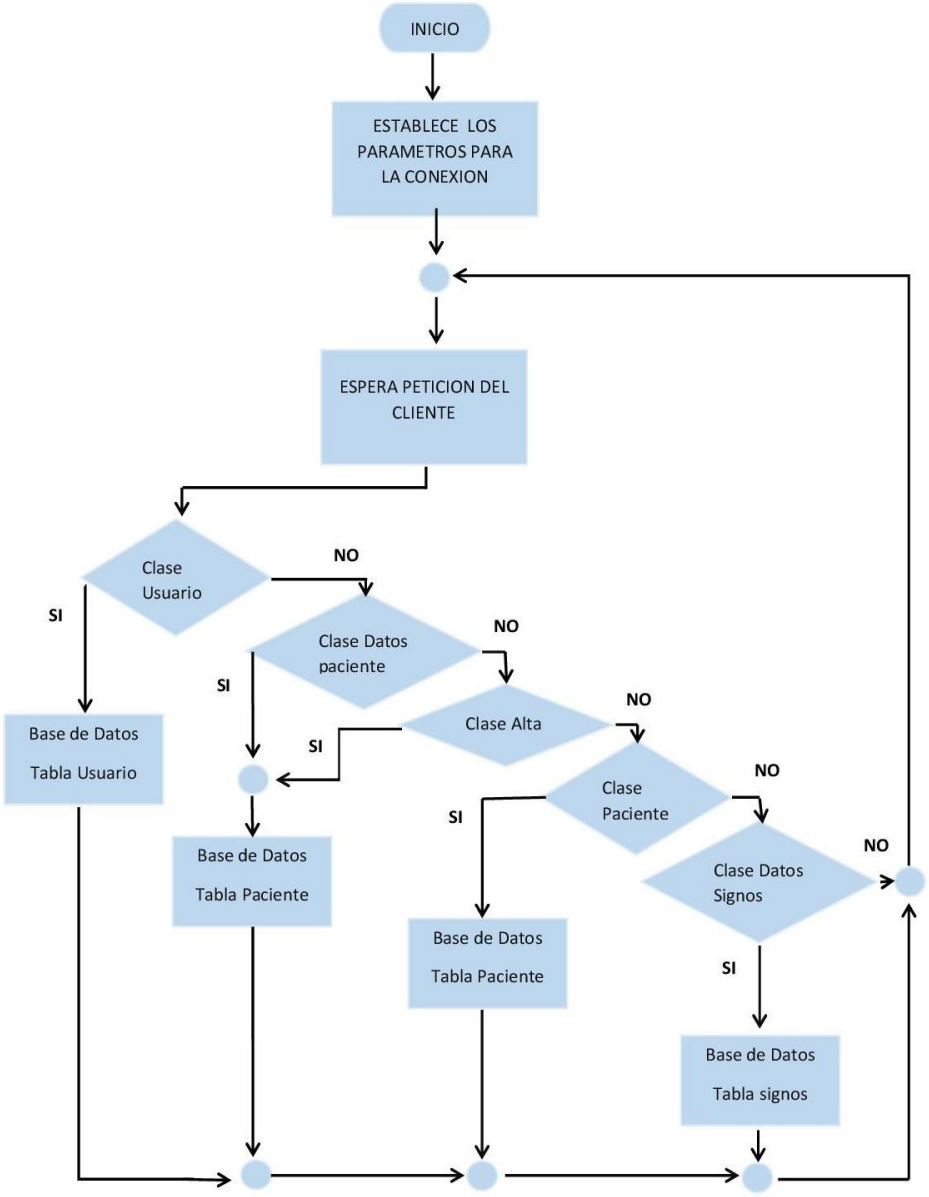


Figura 3.1: Diagrama Servidor.

3.1.4.1 Clase Servidor

Esta clase contiene el método main, el cual es fundamental para el funcionamiento del sistema, aquí se realizan todas las configuraciones necesarias para establecer la comunicación, tales como la configuración del puerto de enlace, siendo este el 12000, así como también, se importan las librerías necesarias que contienen los métodos para la configuración del protocolo XML-RPC y para la extracción de la fecha actual, usando la librería *java.util.date*, que cumple dos propósitos, primero como parámetro de entrada a la base de datos y segundo, para facilitar el registro de la información.

3.1.4.2 Clase Usuario

Para el funcionamiento de esta clase se requiere de dos parámetros de entrada que son el usuario y la contraseña, su propósito es el de autenticar al personal autorizado, éstos parámetros, que deben ser creados previamente y de forma directa en la base de datos, son buscados y comparados con la información registrada lo cual da paso a dos únicas respuestas SI o NO, si el parámetro ingresado no coincide o no existe se genera la segunda respuesta, el mismo que se ve reflejado en un mensaje de error en la pantalla del cliente, caso contrario, si los datos son correctos se genera la primera respuesta y valida al usuario dando paso al siguiente proceso.

3.1.4.3 Clase Paciente

La clase Paciente genera el formulario con la información del paciente y lo etiqueta con un código único aleatorio de ocho dígitos el cual sirve como clave primaria para el almacenamiento en la base de datos, luego invoca a la pantalla monitor y le pasa el código como parámetro.

3.1.4.4 Clase DatoPaciente

Esta clase es invocada desde la pantalla monitor, la cual pasa el código generado en la clase paciente para que haga una búsqueda en la base de datos y muestre el nombre y apellido del paciente así como también la fecha de nacimiento y el doctor a cargo.

3.1.4.5 Clase DatoSignos

Esta clase recibe una línea de código con los siguientes datos: código paciente, SpO2, Temperatura, peso, ECG, frecuencia cardíaca, fecha y alerta, estos datos son almacenados en la tabla Signos mediante un proceso denominado proceso almacenado el cual es un programa interno que se ejecuta dentro de la base de datos lo que permite agilitar la velocidad de inserción de datos, este proceso se realiza con cada uno de los 400 items recibidos.

3.1.4.6 Clase Alta

Cuando se invoca a la clase Alta recibe el código del paciente como parámetro lo que da paso a la modificación del campo Activo que está en la base de datos y lo sobrescribe con un NO, lo cual indica al cliente que el paciente está inactivo y se suspende el envío de información del paciente correspondiente.

3.2. Aplicación de Web

La aplicación es una página web que puede ser accedida desde cualquier navegador que se encuentre dentro de la red LAN, para facilitar el desarrollo de la misma se hace uso de herramientas que agilitan el proceso, tales como: Java Server Faces y Primefaces.

3.2.1 Java Server Faces

La tecnología Java Server Faces 'JSF' es un framework estándar orientado a la interfaz de usuario para la plataforma de Java EE, es decir; es un framework de desarrollo web basado en java. JSF y permite la creación de aplicaciones web sin la necesidad añadir librerías adicionales, ofreciendo herramientas que facilitan el procesos de desarrollo. (Oracle, JavaServer Faces Technology, 2015).

JSF tiene dos funciones principales, la primera es generar una interfaz de usuario cuya respuesta es un HTML que se refleja como una página web que permite apoyar diferentes lenguajes de marcado o entornos de navegadores alternativos como los de los teléfonos inteligentes. La segunda función es la de responder a eventos generados por el usuario invocando a servidores oyentes secundarios, generando otra página web o actualizando la interfaz de usuario, en este sentido se dice que JSF es framework de desarrollo web orientado a eventos. (Oracle, Java Server Faces, 2015).

3.2.2 PrimeFaces

PrimeFaces es una librería de software libre para JSF que ofrece simplicidad y rendimiento ya que , por lo general, la adición de una solución de terceros podría traer una sobrecarga al sistema sin embargo, este no es el caso ya que PrimeFaces no contiene dependencias y nada que configurar debido a su filosofía de mantenerlo lo más ligero posible. Los componentes en PrimeFaces se desarrollan bajo el concepto

de establecer un componente de interfaz de usuario que oculte la complejidad y mantener la flexibilidad y facilidad de uso mientras lo hace, además, posee una fuerte retroalimentación por parte de la comunidad que proporciona nuevas ideas y corrigen los errores. (Prime Tek, 2014) (Civici, 2014).

3.2.3 HTML

Como se mencionó anteriormente, la aplicación web utiliza el Lenguaje de Marcado para Hipertexto, o HTML por sus siglas en inglés, para la construcción y representación visual de la página web que utiliza el protocolo HTTP para la transmisión de datos.

Este lenguaje utiliza marcado de texto para etiquetar los componentes que forman parte de la página web, también permite crear vínculos o enlaces que conectan una página web con otra a través del World Wide Web, además, permite soportar imágenes, sonidos y otros tipos de elementos multimedia. (Consortium W3, 2015).

Para crear un archivo HTML no se requiere de una aplicación compleja sino por el contrario solo se necesita de un editor de texto plano ya que el lenguaje describe la estructura y el contenido semántico del documento web a través de bloques delimitados por los símbolos de *mayor qué* y *menor qué* como por ejemplo <img>, <title>, <p>, , aunque no existe un orden especifico de como estructurar el documento. (Florentino, 2015).

3.2.4 Diseño Pagina Web

La aplicación web está programada en el entorno de desarrollo de Netbeans y está conformada por dos páginas web que son: la página de inicio y la página de signos cuyo funcionamiento se resume en los diagramas de la Figura 3.2 y la Figura 3.3 respectivamente.

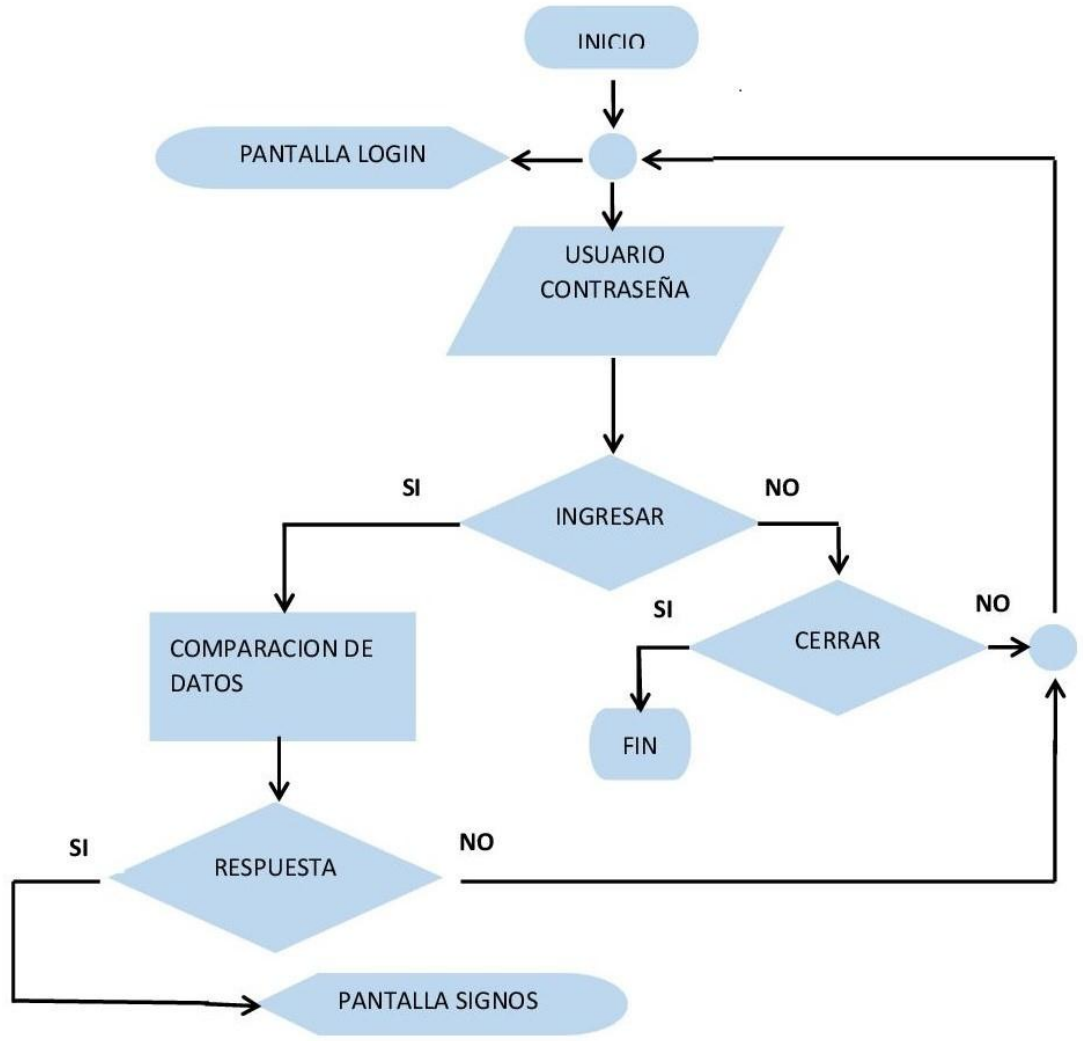


Figura 3.2: Diagrama Página de inicio.

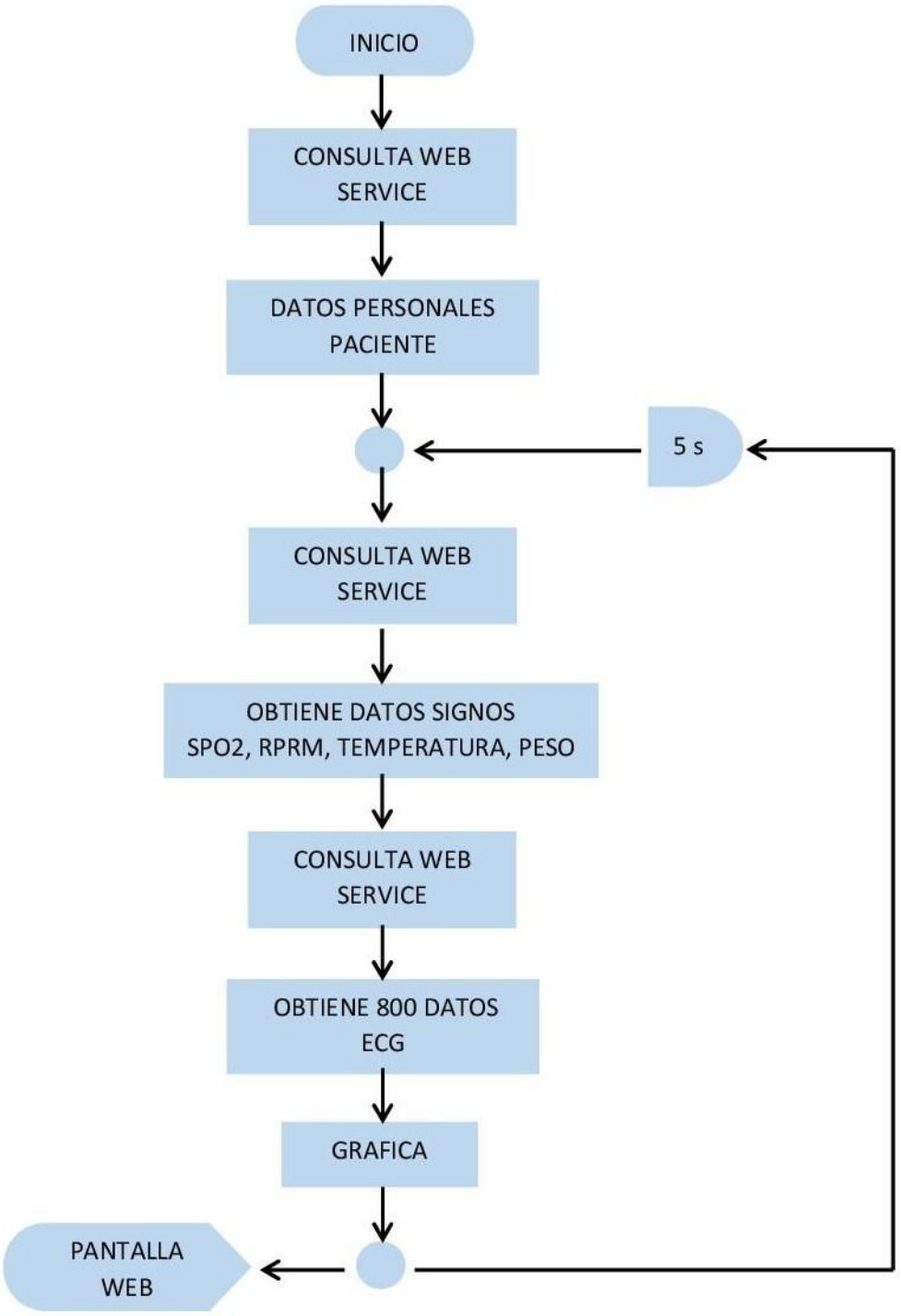


Figura 3.3: Diagrama página de signos.

3.2.4.1 Página Inicial

La página de Inicio o Index se puede apreciar en la Figura 3.4 y está diseñada para que, al igual que el programa elaborado en Python, haga la autenticación del personal mediante el ingreso del usuario y la contraseña que ya han sido generados previamente. La verificación se realiza mediante la conexión con la base de datos mediante las mismas librerías mencionadas anteriormente, donde los campos ingresados son comparados y en caso de haber algún error se despliega un mensaje que alerta al usuario, tal como se ven en la Figura 3.5.

Figura 3.4: Pagina Inicial.

Para agilitar el diseño de la página se utiliza una plantilla que proporciona el JSF, esta plantilla contiene dos secciones, la primera es la cabecera y es donde va el nombre o título de la aplicación, en la segunda sección se encuentra el contenido y para éste caso solo dispone de los dos campos de texto, para ingresar los datos, y sus respectivas etiquetas.

Figura 3.5: Mensaje de error login.

Para realizar el proceso de comunicación con la base de datos se necesita de una clase controlador, denominada entradas.java, esta clase también contiene todas las acciones para realizar las validaciones y la visualización de los mensajes de error, si la consulta es correcta se procede a re direccionar a la siguiente página donde se visualizan los signos vitales del paciente.

3.2.4.2 Página de signos

La página de signos está dividida en tres partes que son: Datos del paciente, Signos del Paciente y Gráfica ECG, cada una manejada por su respectiva clase.

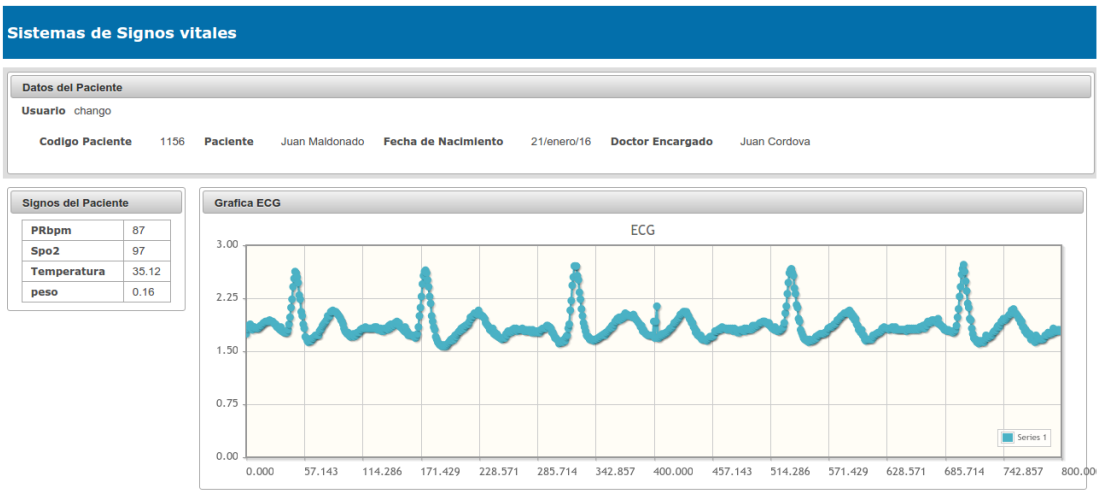


Figura 3.6: Página Web de Signos vitales.

La sección Datos paciente esta manejada por la clase pacSignos.java, la cual devuelve la información del formulario del paciente activo como se ha visto anteriormente, de igual manera, los signos del paciente son extraídos de la base de

datos por medio de la clase SignosDatos.java la misma que consulta el último dato ingresado de la frecuencia cardíaca, saturación de oxígeno, temperatura y peso, esta información es actualizada cada cinco segundos.

Para realizar el gráfico del ECG se utiliza la clase CharView.java, aquí se realiza una consulta a la base de datos para obtener los últimos 800 valores del campo correspondiente, esos datos son almacenados en un arreglo y luego son enviados al componente que realiza la gráfica.

3.3. Base de datos

Como ya se ha mencionado en las secciones anteriores, todo el sistema se conecta con la base de datos para manipular la información que es almacenada de una forma estructurada, denominada tablas, a través del software de gestión de base de datos MySQL que se describe a continuación.

3.3.1 MySQL

Mysql es un sistema de gestión de base de datos relacional (RDBMS) de código abierto y está basado en un lenguaje de consulta estructurado (SQL). Además, posee la ventaja de ser ejecutado en prácticamente cualquier sistema operativo y, a pesar que se puede utilizar en una amplia gama de aplicaciones se lo asocia principalmente con las aplicaciones basadas en la web y la publicación en línea. (Defincion ABC, 2015) (Rouse, 2015).

Una de las características más interesantes de MySQL es que permite recurrir a bases de datos multiusuario a través de la web y mediante diferentes lenguajes de programación, los cuales se adaptan a las distintas necesidades y requisitos. Por otro lado, MySQL es conocida por desarrollar alta velocidad en la búsqueda de datos e información, a diferencia de sistemas anteriores, y se continúa estudiando y desarrollando nuevas versiones que buscan presentar mejoras y avances para permitir un mejor desempeño en toda actividad que requiera el uso de bases de datos relacionales. Entre estas mejoras destacan un nuevo dispositivo de almacenamiento, respaldo para todos los tipos de almacenamientos, réplica segura, planificación de eventos entre otras más. (Defincion ABC, 2015) (Rouse, 2015)

3.3.2 Descripción Base de Datos

La base de datos está formada por tres tablas que son: tabla usuario cuya función es almacenar la información del personal que estará a cargo del paciente, de aquí se extrae los campos Nick, contraseña y nombre para realizar la verificación del usuario cuando se genera una petición desde los clientes o desde la aplicación en Python. La tabla paciente es la encargada de almacenar la información del formulario de cada paciente y la tabla signos crece, de forma dinámica, en función de los datos recibidos que provienen del arreglo enviado desde el procesador central y que contiene los signos vitales, ésta tabla es la más grande de todas pues, almacena un promedio de 500 datos por segundo, de esta manera se garantiza el número necesario de puntos para la gráfica del ECG.

3.4. Desarrollo de aplicación para Clientes

3.4.1 Android Studio

La empresa Google lanzo el IDE oficial para el desarrollo de aplicaciones Android, el software lleva el nombre de Android Studio y, al igual que Java, emplea una programación orientada a objetos. Este IDE está pensado para facilitar el proceso de desarrollo ya que posee una herramienta de empaquetado y etiquetado que permite implementar grandes cantidades de código, además, para aquellos que no desean utilizar el método tradicional de programación, es decir tipiando código, se puede utilizar la herramienta Drag&Drop que permite arrastrar los elementos desde una paleta y realizar una programación más visual. (Google Inc, 2015).

Aunque, lo ideal para revisar el avance de la aplicación es probarla directamente en un dispositivo real, Android Studio dispone de emuladores pre configurados donde se virtualiza un dispositivo con sistema operativo Android, en caso de existir el dispositivo requerido se puede crear uno y configurarlo según las necesidades. Además, tiene un nuevo entorno denominado Google Cloud Messaging que permite enviar datos desde un servidor hacia terminales en la nube (Internet) siendo una forma de enviar notificaciones PUSH para las aplicaciones. (Google Inc, 2015).

3.4.2 Diseño de Aplicación

La aplicación Android funciona de la misma forma que la página web pero, a diferencia de ésta, la gráfica se lo realiza en una tercera pantalla que se describe más adelante, además, se emplea la librería KSoap2 para la comunicación con el servidor. Esta librería sirve para generar el archivo XML que lleva la información que se desea consultar, el proceso se realiza en cada clase a través de los métodos SoapObject, SoapSerializationEnvelope y HttpTransportSE que contienen los parámetros correspondientes.

3.4.2.1 Pantalla y Clase Login

La pantalla Login es la primera que se ejecuta al iniciar la aplicación y presenta el mismo diagrama de la Figura 3.2, está diseñada para que ajuste su tamaño en función del tamaño y tipo de pantalla del dispositivo en cual se ejecuta. Esta pantalla, al igual que en la página web y la aplicación del procesador central, contiene los campos o textbox en los cuales se debe ingresar los datos del usuario, además, para realizar la consulta dispone de un botón llamado INGRESAR, el mismo que realiza la llamada al web service para validar el usuario.

Como es de esperar, detrás del funcionamiento de ésta pantalla existe una clase que la controla y se denomina clase Login, esta clase sirve de controlador y realiza la conexión con el servidor mediante el protocolo SOAP de Java, para que la conexión se establezca, se importar la librería KSoap2, esta librería realiza la conexión con el servidor enviándole los parámetros que se desean consultar a través de un archivo XML que se genera mediante los métodos mencionados.

Figura 3.7: Pantalla Login para aplicación Android.

La clase detecta tres posibles errores que se pueden producir; si los datos ingresados son incorrectos, si no existe conexión a la red o si algún campo no ha sido llenado se genera un mensaje temporal como se muestra en la Figura 3.8 que notifica al usuario cual es la situación, en caso de no existir ningún error se da paso a siguiente pantalla.

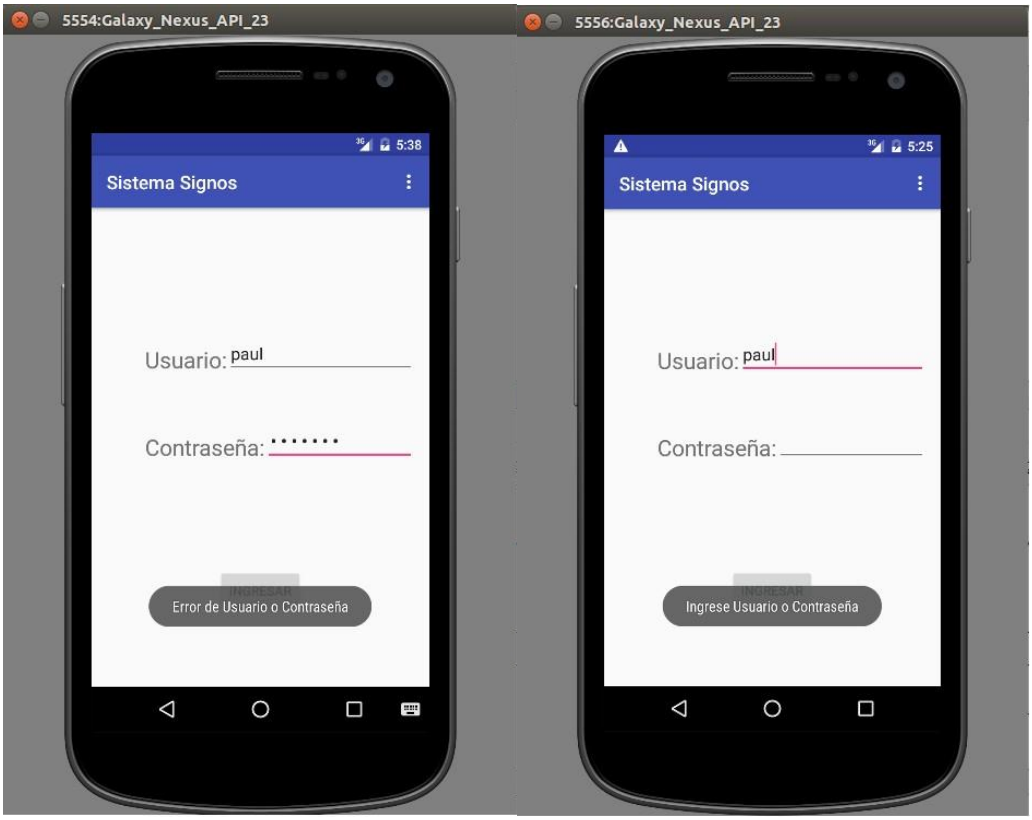


Figura 3.8: Notificaciones de error en pantalla Login.

3.4.2.2 Pantalla y Clase Signos

Esta pantalla equivale a la pantalla monitor del programa en Python y, está pensada en que sea lo más simple posible pero que cumpla con su propósito el cual es mostrar los signos del paciente así como también la información del mismo, además, posee la habilidad de reportar las alarmas que se han generado para mantener al personar siempre informado de la situación actual del paciente. Para cumplir con este propósito, se crea la clase signos en la cual se emplea cuatro etiquetas que indican los parámetros a ser leídos y cada etiqueta se le asigna una variable correspondiente donde se carga el valor obtenido desde la base de datos, esta información, al igual que en la página web, se actualiza cada cinco segundos. El funcionamiento se describe a través del diagrama de la Figura 3.9.

.

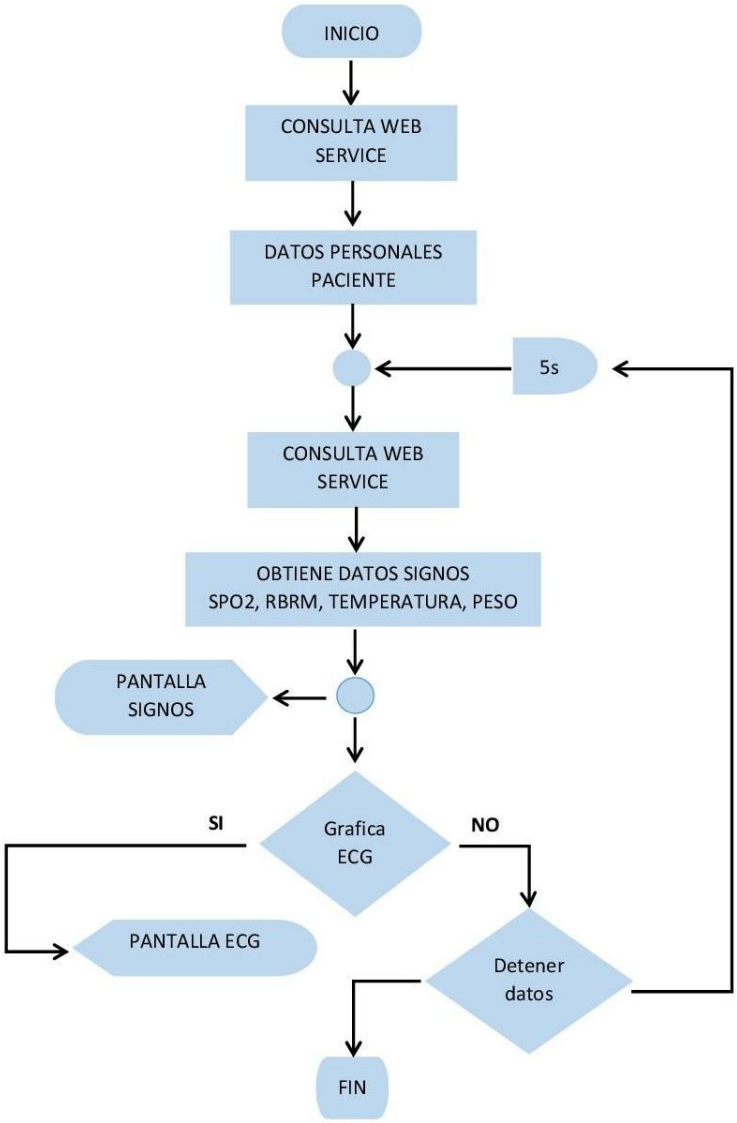


Figura 3.9: Diagrama Pantalla signos para clientes Android.

En la parte inferior de la pantalla están ubicados dos botones, el primer botón denominado PARA DATOS detiene la lectura de la información y el segundo botón, de nombre, GRAFICA ECG llama a otra pantalla donde se visualiza la gráfica que representa el potencial generado por el corazón.

Figura 3.10: Pantalla de signos vitales para clientes Android.

3.4.2.3 Pantalla y Clase ECG

El procedimiento de la pantalla ECG se puede apreciar en el diagrama de la Figura 3.11, donde la gráfica del ECG muestra en tiempo real la actividad cardíaca del paciente, mediante la clase del web service antes mencionada, esta clase envía 800 datos para ser graficados. El proceso se realiza cada 200 milisegundos con el propósito de dar continuidad al gráfico, respecto al paquete anterior de datos, para de esta forma obtener una mejor sensación visual y evitar cortes o pérdida de información.

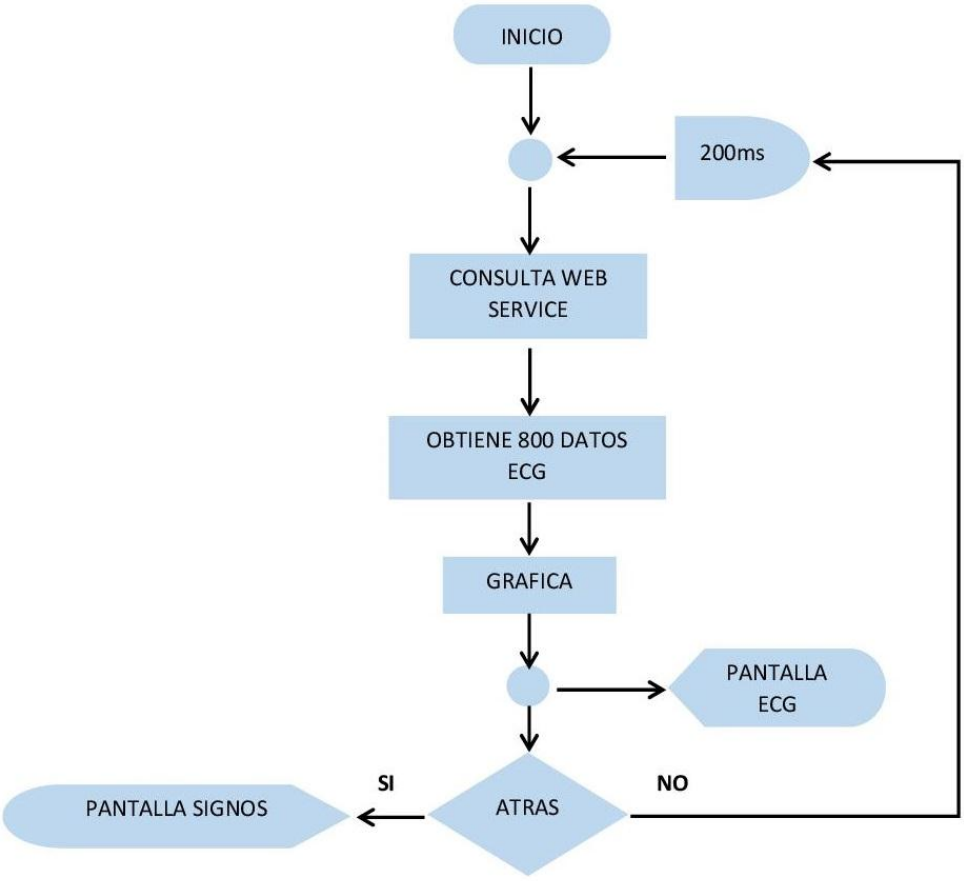


Figura 3.11: Diagrama pantalla ECG para clientes Android.

Para realizar este proceso se crea la clase ECG la cual maneja la librería androiPlot que permite realizar varios tipos de gráficas en el plano XY como se puede observar en la Figura 3.12.

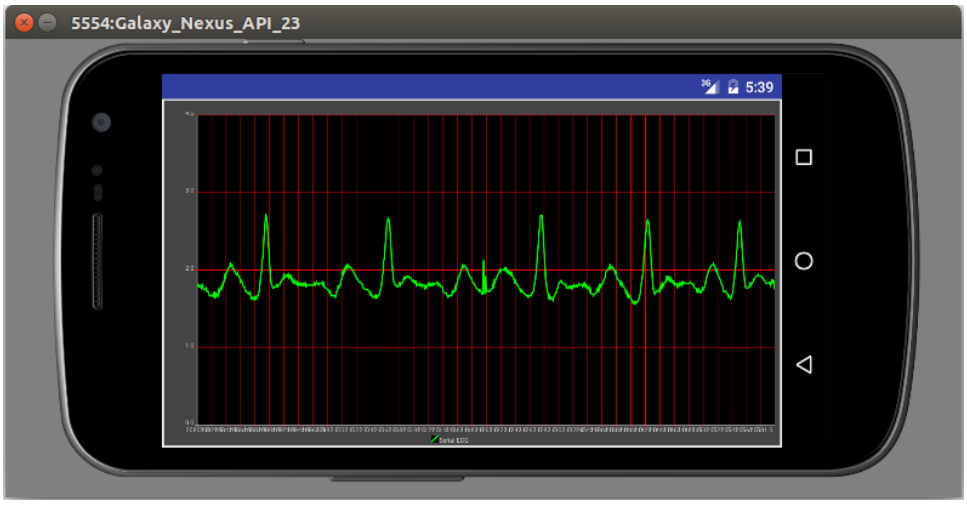


Figura 3.12: Grafica ECG en para clientes Android.

CAPÍTULO 4

ANÁLISIS Y RESULTADOS

El sistema descrito en las páginas anteriores de éste documento tienen la finalidad de aportar, en cierto grado, mecanismos que puedan facilitar el cuidado de pacientes en estado crítico y que requieran una atención constante, empleando los gadgets o dispositivos de uso cotidiano. Aunque, al sistema se lo ha enfocado principalmente al cuidado de los recién nacidos prematuros, debido que la innovación tecnológica no se enfoca en igual proporción a esta etapa crucial en la vida de las personas, también se lo ha orientado a una posible migración hacia otros tipos de pacientes y áreas médicas ya que no está ligado a un hardware específico.

Debido que el objetivo no es crear un equipo médico, la utilidad y eficiencia de este sistema respecto son opiniones personales de quien esté leyendo este documento, es por esta razón, que no se hará mención a esos puntos sino, por el contrario, a continuación se describe los inconvenientes así como también algunas limitaciones que han surgido durante el desarrollo de este proyecto.

4.1. Instrumentación Biomédica

La tarjeta e-Health resulta ser un herramienta sencilla y bastante útil para trabajar con parámetros fisiológicos, lo que permite enfocarse en el procesamiento y manipulación de la información, pero, al ser una plataforma de propósito general no cumple con los requisitos para formar parte de un sistema real, ya que, no posee los filtros necesarios para combatir las interferencias que se encuentran presentes en el medio ambiente y en la red eléctrica. Refiriéndose específicamente a la adquisición del ECG, el mayor inconveniente fue producto del ruido eléctrico, cuya frecuencia es de 60 Hz, que distorsionaba en gran medida la forma de onda, para lo cual, la solución más sencilla es la de implementar un filtro Notch para dicha frecuencia seguido de un filtro pasa bajos de 22Hz, dando como resultado una señal más limpia y legible como se aprecia en la Figura 4.1, sin embargo, el ruido debido al

movimiento sigue afectando la señal, lo que implica que el paciente debe permanecer inmóvil el mayor tiempo posible y, aunque los filtros realicen un buen trabajo el ruido siempre estará presente.

Figura 4.1: Filtrado de la señal ECG.

En lo que se refiere al sensor de pulsioximetría, se presentan valores aproximados, sin embargo, en ocasiones presentan valores dudosos debido a la suciedad del dispositivo y de la piel del paciente por lo que es recomendable realizar una limpieza respectiva previo al uso. Por otra parte, el sensor de temperatura a pesar de ser bastante sensible no es muy preciso cuando el rango de medición varía en un grado centígrado, para mitigar este error, lo más práctico es realizar un promedio de 100 valores evitando de ésta manera que hayan saltos grandes en los puntos decimales, sin embargo, éste promedio representa un retardo pequeño y consumo de recursos para el procesamiento.

4.2. Procesadores

El procesador central original con el que se empezó el proyecto fue el Raspberry B+, este dispositivo presenta un buen desempeño cuando solo se trabaja con los datos de pulsioximetría, temperatura y peso, sin embargo, al graficar el ECG de forma constante y por tiempo prolongado se utilizan muchos recursos del dispositivo lo que

genera un retardo en la presentación de la información, la solución a este problema se hace evidente y se resume en aumentar la capacidad de procesamiento, por lo que se empleó la versión más actual hasta la fecha del dispositivo, el mismo que se describe en el capítulo dos, el tiempo para la visualización de datos se redujo de forma considerable actualizando la gráfica cada tres segundos aproximadamente. Además, en la gráfica se empleó dos buffer de memoria, uno para la recepción y otro para el almacenamiento temporal de datos, hasta que estos son enviados hacia el servidor, de ésta manera se evita la pérdida de información. Para optimizar los recursos, cada proceso se realiza en un hilo independiente.

Otro problema se produjo durante la comunicación entre Arduino y el procesador central, el cual consistía en que él envío de información hacia el procesador central era más rápida que el proceso de recepción y clasificación de datos, lo que implica un desfase en el orden de llegada de la información dando como resultado una mala interpretación de los datos, para solucionar este problema se realizó una sincronización mediante un proceso de petición y respuesta, para eso se envía una letra que al ser interpretada por el Arduino da paso a la lectura la información proveniente de los sensores y los envía como una trama de 5 bytes.

4.3. Servidor y clientes

En cuanto al servidor no se presentaron mayores inconvenientes, solo se debe tener presente algunas consideraciones como; en lo que respecta al ingreso directo de la información en la base de datos es muy lento, por lo que se aconseja utilizar el proceso almacenado que optimiza el tiempo de ingreso de datos hacia la tabla correspondiente. Otra consideración, es que al estar trabajando con lenguajes de programación distintos se debe utilizar protocolos de comunicación compatibles, como son el XLMRPC para el servidor y cliente web, así como el protocolo SOAP para el cliente Android los mismos que trabajan con el formato XML para el manejo de la información.

4.4. Estructura

Para dar un enfoque más real al sistema, se realizó la estructura de una termo-cuna para demostrar el funcionamiento del mismo, para ello, se utilizó un perfil de hierro

en forma de C, cuyo espesor es de 1.5 mm. La estructura está dividida en tres partes desarmables que son: base, columna principal y soporte para la cuna.

La base está construida con tres secciones del perfil, las mismas que están soldadas entre ellas en forma de H, en esta parte reposa el peso restante de la estructura junto con sus aditamentos, para facilitar el desplazamiento de la estructura se dispone de cuatro ruedas, una en cada extremo, cada rueda incluye un freno para evitar el movimiento involuntario, además, la base dispone de una ranura donde encaja la columna principal, la cual se sujeta mediante pernos.

Figura 4.2: Base de la estructura.

La columna principal se constituye de dos secciones de 180 cm y 65 cm soldadas de manera perpendicular la una respecto a la otra, formando una L invertida. En la parte superior, siendo la más corta, se sujeta el calefactor y al mismo que se lo reviste con una lámina de aluminio que también ayuda a concentrar el calor hacia abajo, en la otra sección se sitúa la pantalla monitor y a una distancia 80 cm desde la parte superior se fija el soporte para la cuna.

Figura 4.3: Columna principal con soporte para cuna.

El soporte para la cuna es un sección de 60cm, en cuyo extremo está soldado un ángulo con dos orificios para sujetarse a la columna mediante pernos, para ayudar a distribuir el peso de mejor manera se aplican dos pie de amigo debajo del soporte. Para la construcción de la cuna se utilizó madera MDF de 6 mm y acrílico de 4 mm de espesor para la paredes abatibles, la base de la cuna consiste en un rectángulo de 45x40 cm en cuyos bordes se adhieren con pegamento paredes de 5 cm de alto hasta formar un cajón, de forma similar, se genera otro cajón interno a 4 mm de separación respecto al otro, el espacio entre paredes sirve de ranura donde encajan las paredes de acrílico que resguardan al paciente. En el cajón formado se ubica la balanza y encima de ésta se coloca una plataforma, donde descansará el paciente. En la parte inferior de la cuna, se extienden dos paredes laterales que sirven para fijar los rieles que facilitan el desplazamiento de otra plataforma que soportará el teclado y mouse necesarios para interactuar con el sistema.

Figura 4.4: Soporte y cuna.

Con el propósito de obtener un mejor aspecto, cada parte es pintada con pintura de esmalte color blanco hueso.

Figura 4.5: Estructura pintada.

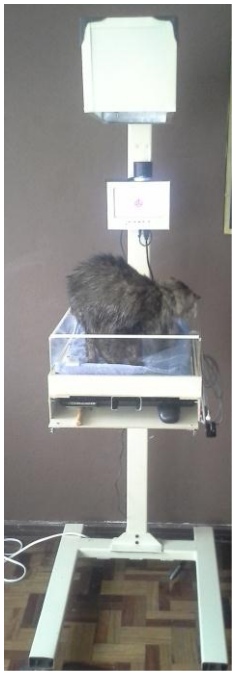


Figura 4.6: Estructura finalizada.

Para mejorar la estabilidad de la estructura, se recomienda usar un perfil de mayor espesor o en su defecto reforzar la partes más endebles de la misma.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones:

- Lejos de presentar un equipo médico, el sistema propuesto plantea la idea para generar una posible alternativa en la implementación de otros mecanismos que ayuden a mejorar la atención de pacientes de alto riesgo mediante el uso de hardware y software libre y, aunque eso implique ciertas limitaciones, tales como: susceptibilidad a interferencias, oscilación de mediciones, capacidad de procesamiento y ralentizaciones, se puede mitigar estos inconvenientes, en cierto grado, mediante el uso adecuado de las herramientas de programación y el desarrollo de hardware adicional para obtener rangos más permisibles.

- El sistema además de generar señales de alerta al médico residente y enfermeros de forma audible, también presenta una alarma al médico tratante de forma remota dentro del ambiente hospitalario; mejorando la respuesta a posibles cambios en la medicación o emergencias ya que el tiempo de latencia es de 2 segundos.

Recomendaciones:

Como recomendaciones finales, se debe tener presente algunos puntos clave que influyen directamente en el funcionamiento del sistema, tales como:

- Utilizar un equipo dedicado, de preferencia de última generación, para ejecutar el servidor web, de esta forma se optimiza la velocidad de almacenamiento de la información dentro de la base de datos.

- Debido a las limitaciones que presentan los mini ordenadores, Smartphones y tablets respecto a un equipo informático de alto rendimiento, es recomendable utilizar hilos independientes para todas las etapas en las que se manipula la información, es decir; recepción, envío, visualización de datos, etc. De esta manera el procesador efectúa varias tareas al mismo tiempo en lugar de una tras otra, evitando demoras innecesarias.

- Para evitar llamar al servidor constantemente y saturarlo cada vez que se envía un dato, es mejor enviar la información hacia el servidor en forma de paquetes que contienen cierta cantidad de datos.

- En lo que respecta al hardware adicional, es mejor utilizar componentes de precisión y que sean inmunes al ruido para no contaminar las señales adquiridas. Así como también, utilizar cables apantallados y planos de tierra, en los circuitos impresos, para disminuir las interferencias.

BIBLIOGRAFÍA

Anliker, U. W., Lukowicz, P., Tröster, G., Dolveck, F., Baer, M., Keita, F., . . . others. (2004). AMON: a wearable multiparameter medical monitoring and alert system. Information Technology in Biomedicine, IEEE Transactions on, 8(4). 415-427.

Arduino. (2015). *ARDUINO. Arduino UNO & Genuino UNO*. Recuperado el 30 de Junio de 2015, de https://www.arduino.cc/en/Main/ArduinoBoardUno

Chen, W., Nguyen, S. T., Coops, R., Oetomo, S. B., & Feijs, L. (2009). Wireless transmission design for health monitoring at neonatal intensive care units. *Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009. 2nd International Symposium on*, 1-6. IEEE.

Ciani, O., Piccini, L., Parini, S., Rullo, A., Bagnoli, F., Marti, P., & Andreoni, G. (2008). Pervasive technology in Neonatal Intensive Care Unit: a prototype for newborns unobtrusive monitoring. *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, 1292-1295. IEEE.

Civici, C. (2014). *PrimeFaces User Guide 5.0.* Recuperado el 3 de Noviembre de 2015, de https://speakerd.s3.amazonaws.com/presentations/9d21c4d0bcc50131faee127b1bf32aaa/primefaces_user_guide_5_0.pdf

Consortium W3. (2015). *What is HTML?* Recuperado el 3 de Noviembre de 2015, de https://www.w3.org/html/

Defincion ABC. (2015). *Definicion de MySQL*. Recuperado el 4 de Noviembre de 2015, de http://www.definicionabc.com/tecnologia/mysql.php

Egan, F. (2006). El ABC del cuidado de enfermería en los bebés prematuros extremos–Segunda parte–. *Rev Enfer Neon*. Recuperado el 10 de Marzo de 2015, de www. fundasamin. org. ar/download. php

Eklund, J. M., McGregor, C., & Smith, K. P. (Agosto de 2008). A method for physiological data transmission and archiving to support the service of critical care using DICOM and HL7. *En Engineering in MedicineEngineering in Medicine and Biology Society, 2008. EMBS 2008.30th Annual International Conference of the IEEE*, 1486-1489. IEEE.

Florentino, M. (2015). *Mozilla, Developer Nerwork*. Recuperado el 3 de Noviembre de 2015, de https://developer.mozilla.org/es/docs/Web/HTML

Google Inc. (2015). *Meet Android Studio*. Recuperado el 28 de Noviembre de 2015, de https://developer.android.com/studio/intro/index.html?hl=es#project-structure

Gordillo Padilla, F. X. (2014). Diseño e implementación de un sistema de monitoreo remoto de señales biológicas, electrocardiograma, oxímetro de pulso, tensiómetro digital (Tesis de Titulación). 7-14. Cuenca, Ecuador. Obtenido de http://dspace.uazuay.edu.ec/handle/datos/4248

Hacks, C. (2013). *e-Health Sensor Platform V2.0 for Arduino and Raspberry Pi*. Recuperado el 16 de Junio de 2015, de https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platformarduino-raspberry-pi-medical

Microchip, T. (2011). *Charge Pump DC-to-DC Voltage Converter*. Recuperado el 20 de Enero de 2015, de http://ww1.microchip.com/downloads/en/DeviceDoc/21465C.pdf

Mora, J., RUELAS, E., & VELAZQUEZ, A. (2004). Guía Tecnológica No. 2: Cuna de calor radiante. Recuperado el 28 de Abril de 2015, de http://www.cenetec.salud.gob.mx/descargas/biomedica/guias_tecnologicas/2gt_cuna.pdf

Morilla, G. A. (2006). Historia de la incubadora. *Red de salud de Cuba, Cuba*, 1-6. Recuperado el 4 de Marzo de 2015, de http://www.sld.cu/galerias/pdf/sitios/pediatria/historia_de_la_incubadora.pdf

Olimex, C., & MCI, E. (2015). *Celda de Carga - 50Kg*. Recuperado el 25 de Julio de 2015, de http://www.olimex.cl/shop/product/celda-de-carga-50kg-10259?category=248

Oracle, C. (2015). *Bienvenido a NetBeans y www.netbeans.org*. Recuperado el 10 de Octubre de 2015, de https://netbeans.org/index_es.html,

Oracle, C. (2015). *Java Server Faces*. Recuperado el 20 de Octubre de 2015, de http://www.javaserverfaces.org

Oracle, C. (2015). *JavaServer Faces Technology*. Recuperado el 20 de Octubre de 2015, de http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html

Oracle, C. (2015). *JavaServer Faces Technology Overview*. Recuperado el 5 de Octubre de 2015, de http://www.oracle.com/technetwork/java/javaee/overview-140548.html

Oracle, C. (2015). *NetBeans IDE Features*. Recuperado el 10 de Octubre de 2015, de https://netbeans.org/features/index.html

Pelaez, R. G. (2014). *Diseño e implementación de un sistema de un control electrónico de cuna de calor radiante para la termorregulación neonatal.* Recuperado el 9 de Abril de 2015, de http://148.206.53.84/tesiuami/UAMI15902.pdf

Prime Tek. (2014). *Why PrimeFaces*. Recuperado el 3 de Noviembre de 2015, de http://www.primefaces.org/whyprimefaces

Raspberry, P. F. (2015). *What is a Raspberry Pi?* Recuperado el 8 de Julio de 2015, de https://www.raspberrypi.org/help/what-is-a-raspberry-pi/

Rouse, M. (2015). *TechTarget.* Recuperado el 9 de Noviembre de 2015, de Definicion MySQL: http://searchdatacenter.techtarget.com/es/definicion/MySQL

Ruiz Ibáñez, C., Zuluaga De Cadena, Á., & Trujillo Zea, A. (2007). Telemedicina: introducción, aplicación y principios de desarrollo.

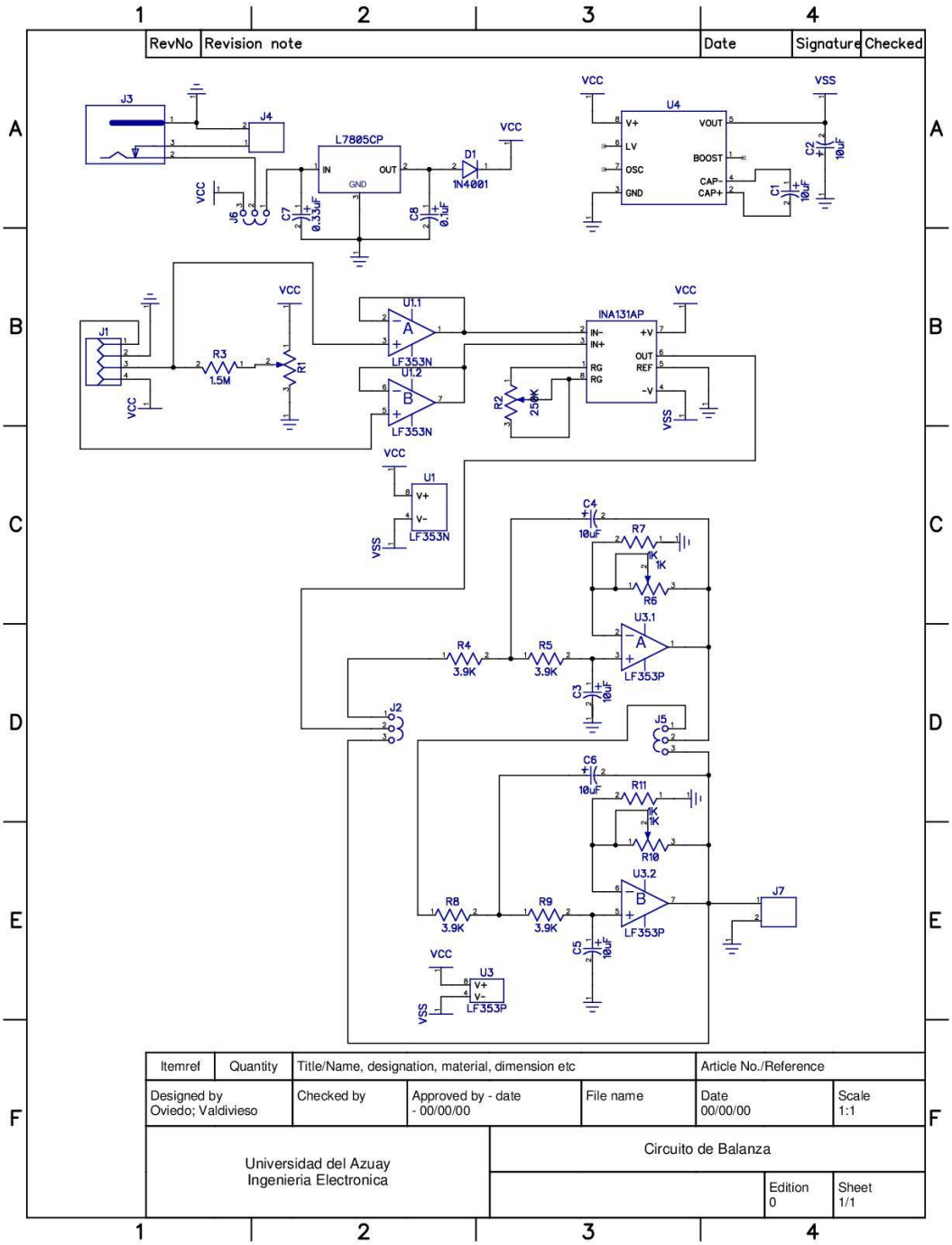
Sánchez, J. E. (2007). *Circuitos de Acondicionamiento Para Sensores Resistivos.* Obtenido de https://es.scribd.com/doc/43054414/7-Circuitos-de-Acondicionamiento-Para-Sensores-Resistivos

Scripting, N. I., & Userland, S. I. (2011). *Simple cross-platform distributed computing, based on the standards of the Internet.* Recuperado el 6 de Octubre de 2015, de http://www.xmlrpc.com/

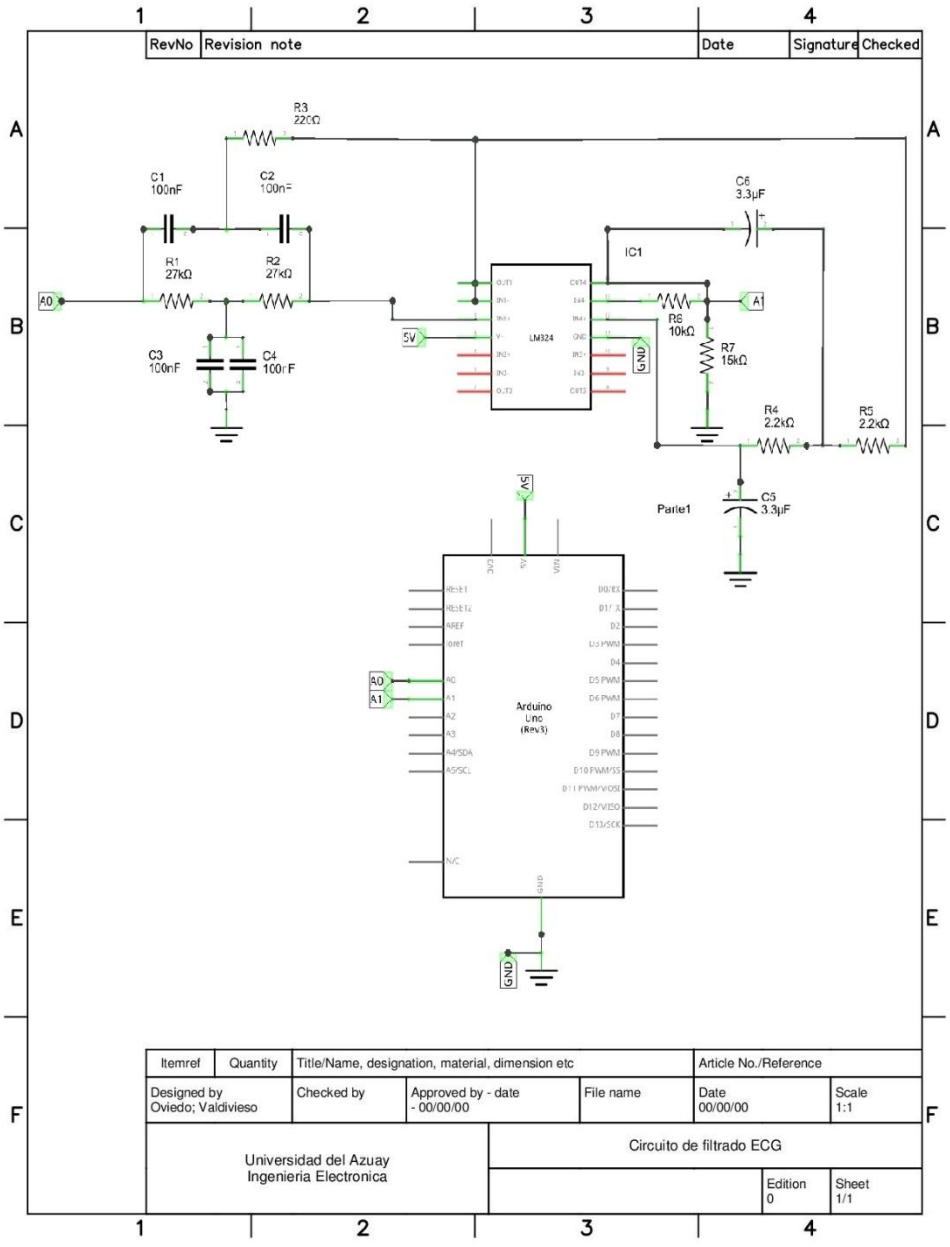
Upton, E. (2 de Febrero de 2015). *Raspberry Pi 2 on sale*. Recuperado el 8 de Julio de 2015, de https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/

ANEXOS

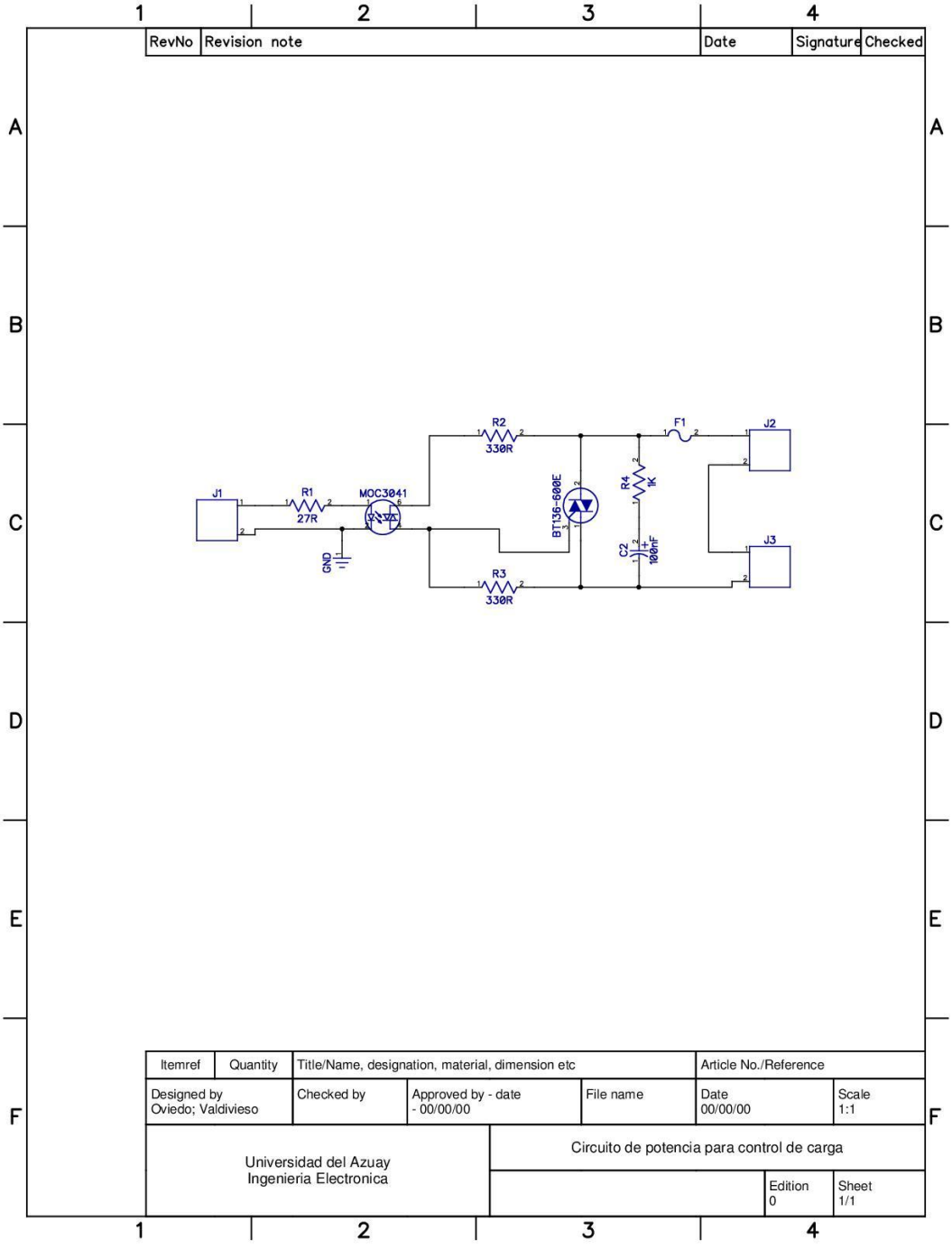
Anexo 1: Circuito de balanza



Anexo 2: Circuito de filtrado para ECG



Anexo 3: Circuito de potencia para control de carga



Anexo 4: Código para configuración PID

```
#include <eHealth.h>
#include <PinChangeInt.h>
#include <PID_v1.h>
double Setpoint, Input,Output;
int PWMpin =3; // LED in series with 470 ohm resistor on pin 10
double Kp=2, Ki=5, Kd=1;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);
void setup()
{
  // no setup needed
  // TCCR1B = TCCR1B & B11111000 | B00000101;
     TCCR2B = TCCR2B & B11111000 | B00000111;
     //Serial.begin(115200);
     Setpoint = 36;
  //Enciende el PID
  myPID.SetMode(AUTOMATIC);
  myPID.SetOutputLimits(0,255);
}
void loop()
{
      delay(100);
      float temperature = eHealth.getTemperature();
      Input = temperature;
      myPID.Compute();
      analogWrite(PWMpin, Output);
}
```

Anexo 5: Código de configuración para Arduino 1

```
// Importación de librerías
#include <eHealth.h>
#include <PinChangeInt.h>
// Declaración de variables
int ECG;
int Temp_pacient;
int Peso;
int Bpm;
int Spo2;
byte serialByte;
int cont = 0;
//Configuracion
void setup(){
  Serial.begin(115200);
  eHealth.initPulsioximeter();
  PCintPort::attachInterrupt(6, readPulsioximeter, RISING);
 }
// Lazo principal
void loop(){
  while(Serial.available()>0){
    serialByte = Serial.read();
    if (serialByte == 't'){
     ECG = analogRead(A1)/4;
      Temp_pacient = analogRead(A3)/4;
      Peso = analogRead(A5)/4;
      Bpm = eHealth.getBPM();
      Spo2 = eHealth.getOxygenSaturation();
      Serial.write(ECG);
      Serial.write(Temp_pacient);
      Serial.write(Peso);
```

```
      Serial.write(Bpm);
      Serial.write(Spo2);
      }
  }
}
//Funciones adicionales
void readPulsioximeter(){
  cont ++;
  if (cont == 50) {
    eHealth.readPulsioximeter();
    cont = 0;
  }
}
```