



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA DE  
VALPARAÍSO



**Esteban Enrique Pérez Palacios**

# Dispositivo portátil para toma de exámenes médicos con prestaciones en Telemedicina

Informe Proyecto de Título de Ingeniero Civil Electrónico



**Escuela de Ingeniería Eléctrica  
Facultad de Ingeniería**

Valparaíso, 27 de febrero de 2018



# Dispositivo portátil para toma de exámenes médicos con prestaciones en Telemedicina

Esteban Enrique Pérez Palacios

Informe Final para optar al título de Ingeniero Civil Electrónico,  
aprobada por la comisión de la  
Escuela de Ingeniería Eléctrica de la  
Facultad de Ingeniería de la  
Pontificia Universidad Católica de Valparaíso  
conformada por

Sr. Francisco Javier Alonso Villalobos

Profesor Guía

Sr. David Velasco

Segundo Revisor

Sr. Sebastián Fingerhuth

Secretario Académico

Valparaíso, 27 de febrero de 2018

*A mis abuelos Melanio Palacios Briones, Eugenia Espinoza Tello y Benito Pérez Fajardo, Q.E.P.D.  
A mis padres Leopoldo Pérez Pérez y Julieta Palacios. A mi hermano Alexander Pérez Palacios.*

# Agradecimientos

Este presente proyecto no habría sido posible de realizar sin la ayuda y consejos de múltiples personas, a quienes me gustaría expresar mi más profundo agradecimiento.

A Felipe Delgado Rodríguez por la paciencia y comprensión, al enseñarme y ayudarme en las mejoras de diseño y creación de base de datos del dispositivo final. A Javier Castro Dutra por ser mi compañero de estudio, guiarme y acompañarme durante todos los años de estudio, también a su familia que fueron un gran apoyo para mí. A Gastón Bizama Escobar, por estar conmigo los primeros años que fueron los más difíciles y nunca distanciarse de mí. A Nicolás Carmona Olivares por ser un gran amigo, enseñarme de electrónica y robótica, y ayudarme en el diseño y confección del dispositivo final. A Giovanni Campos Hernández y Francisca Zarate Muñoz, que durante mi último año estudiantil fueron más que mis compañeros y amigos, se transformaron en una familia.

Por ultimo quiero agradecer a mis abuelos Melanio Palacios Briones, Eugenia Espinoza Tello y Benito Pérez Fajardo, Q.E.P.D. A mis padres Leopoldo Pérez Pérez y Julieta Palacios Espinoza. A mi hermano Alexander Pérez Palacios. Por ser mi compañía, mi fortaleza, mi esperanza, mi guía y mi apoyo incondicional.

*Valparaíso, 27 de febrero de 2018*

Esteban Pérez

# Resumen

Dentro de las diferentes problemáticas que presenta el sistema de salud nacional, se abordó como una solución efectiva y a largo plazo la incorporación de nuevas tecnologías que permiten aplicar la telemedicina, como una solución válida para mejorar las prestaciones de salud en el país. Se vio cuál es su desarrollo en Chile y el mundo, además la forma en que este proyecto puede ayudar a su incorporación por medio de la aplicación de la ingeniería y especialmente de la electrónica.

En base a un avance previo, se buscó la forma de llegar a un dispositivo final que cumpliera con todos los requerimientos que planteaba este proyecto en sus inicios. Se creó un hardware ergonómico y transportable, que permite llevar el dispositivo a cualquier zona donde se quiera realizar la toma de exámenes. También se modificó, el uso y funcionamiento del software, llegando a una interfaz estable y cómoda de usar. Con ello se obtuvo el dispositivo final que se fijó como objetivo principal del proyecto.

Se realizó la incorporación de una base de datos como nueva variante en el proyecto, esta puede almacenar los datos de todas las mediciones realizadas, además de la información de médicos a cargo y pacientes tratados. De esta forma se puede realizar un seguimiento a las personas que poseen enfermedades crónicas como diabetes o asma, y también se pueden visualizar estos resultados a través de internet.

Luego se analizaron los resultados obtenidos por el dispositivo, realizando una comparación entre la máquina médica y los sensores con los que disponía el consultorio de la localidad. Se muestra paso a paso la comparación y la diferencia entre los datos que entrega el dispositivo portátil y los sensores médicos. Verificando la efectividad del proyecto mediante el cálculo de los errores en cada caso.

Finalmente se llegó a un dispositivo cómodo y versátil, que recibió la aprobación de paramédicos y enfermeros, para incorporarlo como una óptima solución que permita enfrentar las diferentes falencias del sistema de salud nacional.

Palabras claves: Dispositivo portátil, E-Health, sensores, salud, telemedicina, interfaz gráfica, hardware, software, Raspberry Pi.

# Abstract

Between diverse problematic that presents the health national system, it was address how an effective solution and the incorporation of a long period of new technologies that allow to applicate the telemedicine, how a validate solution in order to improve the perks of the health in the country. It was seen the development in Chile and the world, besides the way that this project can help to the incorporation through the application of the engineering and specially the electronic one

In a base of a previous advance it was search the way to reach a final device that fulfil with all of the requirement that this project lay out since it begins. It was created an ergonomic hardware and transportable, that allows to carry the device anywhere where it is need to take exams. Also it was modify, the use and running of the software, becoming to a stable interface and comfortable way to use. With this it was achieve the final device that it was set as the main goal of the project.

It was conduct the incorporation of a data base as a new variant in the project, this one can handle the data of all of the realize measuring, furthermore the information of the doctor in charge and the treated patient. In that way it can be made a tracking to the people that have a chronic illness such as diabetes and asthma. In addition the results can be view through the internet.

After the results are obtain via the device, doing a comparison between the medical machine and the sensors that belong to the medical facility of the town. It is show step by step the comparison and the difference between the data that is submitted by the portable device and the medical sensors. Checking the effectively of our project by the calculus of the mistakes in each case.

Finally it was reach a versatile and comfortable device, which receive the endorsement of paramedics and nurses, to incorporate it as an optimal solution that allow to deal with misconceptions of the health national system.

Key words: Portable device, E-Health, sensors, health, telemedicine, graphic interface, hardware, software, Raspberry Pi.

# Índice general

Introducción.....	1
1 Antecedentes del proyecto.....	4
1.1 Problemática .....	4
1.2 Estado del arte .....	5
1.2.1 Telemedicina .....	5
1.2.2 Telemedicina a nivel mundial y nacional .....	6
1.2.3 Equipos actuales para telemedicina .....	7
1.3 Solución propuesta.....	9
1.3.1 Plataforma programable.....	9
1.3.2 Plataforma médica .....	10
1.3.3 Ensamblaje de Plataforma E-Health con Raspberry Pi.....	12
1.3.4 Sensores y comandos para obtención de datos en consola .....	12
1.4 Estado de avance.....	19
1.4.1 Software del sistema.....	19
1.4.2 Hardware del sistema.....	24
2 Consideraciones técnicas para mejorar el dispositivo .....	28
2.1 Protocolos de comunicación .....	28
2.1.1 UART (Transmisor y Receptor Asíncrono Universal).....	28
2.1.2 USB (Bus serial universal).....	29
2.1.3 I2C (Circuito inter-integrado) .....	29
2.1.4 SPI (Interfaz serial periférica) .....	30
2.2 Estudio y análisis de Raspberry Pi .....	30
2.3 Adquisición de pantalla.....	33
2.3.1 Adafruit HDMI 5" 800x480 Display Backpack - With Touchscreen .....	33
2.3.2 Pi Foundation Display 7" Touchscreen .....	34
2.4 Adquisición de modulo 3G.....	34
2.4.1 Módulo 3G / GPRS, SIM5215, adaptado por Cooking-Hacks.....	35
2.4.2 Dongle Huawei E303 .....	35
3 Creación y armado del hardware para el sistema .....	37

---

3.1 Propuesta de diseño .....	37
3.2 Diseños e impresiones de piezas para la maleta .....	38
3.2.1 Soporte pantalla táctil .....	38
3.2.2 Banco de baterías .....	39
3.2.3 Caja para plataforma E-Health.....	40
3.3 Resultado final.....	41
3.4 Calibración de sensores gráficos .....	42
<b>4 Modificación y actualización de software.....</b>	<b>45</b>
4.1 Proceso de desarrollo de la interfaz gráfica .....	45
4.1.1 ¿Qué es una interfaz gráfica? .....	45
4.1.2 Lenguaje utilizado .....	46
4.2 Modificaciones a la interfaz gráfica.....	46
4.2.1 Implementaciones nuevas .....	46
4.2.2 Comparación de las modificaciones realizadas .....	48
<b>5 Creación e implementación de una base de datos .....</b>	<b>54</b>
5.1.1 ¿Qué hacen los programas utilizados?.....	54
5.1.2 Instalación de los programas .....	55
5.2 Programación y funcionamiento de la base de datos.....	56
5.3 Resultado y visualización de la base de datos en la interfaz gráfica.....	59
5.3.1 Visualización gráfica de la nueva incorporación.....	59
<b>6 Resultados finales y comparación de mediciones .....</b>	<b>62</b>
6.1 Comparación de exámenes realizados .....	62
6.1.1 Temperatura corporal.....	62
6.1.2 Medición de pulso y oxígeno en la sangre.....	63
6.1.3 Medición de presión arterial .....	65
6.1.4 Medición de glucosa .....	66
6.1.5 Realización del electrocardiograma .....	67
6.1.6 Medición de flujo respiratorio.....	68
6.1.7 Medición de un electromiograma .....	69
6.2 Cálculo de errores en diferentes mediciones .....	70
6.2.1 Tabla de resultados .....	70
<b>Discusión y conclusiones.....</b>	<b>73</b>
<b>Bibliografía .....</b>	<b>76</b>
<b>A Manual de uso.....</b>	<b>79</b>
A.1 Encendido de la plataforma .....	79
A.1.1 Conexión sensor de pulso y oxígeno en la sangre .....	80
A.1.2 Conexión sensor de flujo respiratorio .....	81
A.1.3 Conexión sensor de glucosa.....	82



---

A.1.4 Conexión sensor de presión arterial.....	82
A.1.5 Conexión sensor de temperatura .....	83
A.1.6 Conexión sensor para electromiograma.....	83
A.1.7 Conexión sensor para electrocardiograma.....	84
A.2 Como solucionar inconvenientes .....	84
A.2.1 Caso de perdida en la lectura de los sensores .....	84
A.2.2 Caso de falla en todo el sistema.....	85
<b>B Programación dispositivo portátil.....</b>	<b>86</b>
B.1 Programación sensores biométricos .....	86
B.1.1 Librería de programación (eHealth.cpp) .....	86
B.1.2 Sensor de temperatura (temperatura.cpp) .....	106
B.1.3 Sensor de pulso y oxígeno (saturometro.cpp) .....	107
B.1.4 Sensor de flujo respiratorio (respiracion.cpp).....	108
B.1.5 Electrocardiograma (ecg.cpp).....	109
B.1.6 Electromiograma (emg.cpp) .....	109
B.1.7 Glucómetro (glucometro.cpp) .....	110
B.1.8 Esfigmomanómetro (presion.cpp) .....	111
B.2 Programación interfaz gráfica .....	111
B.2.1 Envío de datos via mail (a.php) .....	112
B.2.2 Eliminación de mediciones (borrar.php) .....	113
B.2.3 Introducción de datos del paciente (datos.php) .....	115
B.2.4 Guardado de las mediciones (datos2.php) .....	119
B.2.5 Ventana gráficas (graficas.php) .....	121
B.2.6 Ventana de obervaciones (observacion.php) .....	123
B.2.7 Configuración y creación de PDF (pdf.php) .....	126
B.2.8 Ventana principal de inicio de sesión (pincipal.php) .....	131
B.2.9 Creación del reporte (reporte.php) .....	133
B.2.10 Ventana de menú principal (p.html).....	135
6.2.2 Datos almacenados en el sistema (resultados.php) .....	135
B.3 Programación y conexión a la base de datos .....	140
B.3.1 Base de datos (ehealth_bd.sql) .....	140
B.3.2 Conexión entre la base de datos y la interfaz gráfica (conexión_bd.php) .....	142

# Introducción

El avance en la tecnología es cada vez más rápido y la gran mayoría de las problemáticas se enfocan a un mundo interconectado, tal como lo hace la telemedicina, la cual tiene como objetivo principal facilitar la atención médica en diferentes áreas de la medicina, ya sea en la toma de exámenes como en la atención directa con un médico. Esto permite ampliar la cobertura de los servicios médicos, incluyendo a las zonas más aisladas, con ello se consigue la equidad en el sistema de salud del país.

A nivel mundial, la telemedicina está en fuerte desarrollo, regiones como América Latina, Asia y África buscan alcanzar la implementación completa de este tipo de servicios, tal como ya lo han hecho Estados Unidos y el Reino Unido. Chile se posiciona en las redes internacionales de la telemedicina y esto lo ha logrado gracias a la creación de centros de investigación en la Universidad de Chile. Esta cuenta con un reciente proyecto que busca conectar los equipos médicos de salud a las zonas más alejadas, a fin de mejorar la atención de los pacientes, democratizar el conocimiento y el acceso a la salud.

Es con la necesidad de interconexión que cada día se acrecienta más, que diversas empresas de tecnología y electrónica han desarrollado sensores y plataformas médicas que están directamente orientadas a la telemedicina. Existen empresas que venden una diversa gama de productos industriales que se comercializan a nivel mundial y que cumplen funciones que son de suma relevancia para aplicar la telesalud en los países. Sin embargo, existen otras empresas que desarrollan plataformas de código abierto, que permiten la elección y edición de funciones médicas con el fin de que el usuario adapte las prestaciones de estos dispositivos según sus necesidades.

La plataforma E-Health diseñada por Cooking Hacks con el objetivo de ir en ayuda de investigadores y desarrolladores para medir datos de sensores biométricos con fines de experimentación y de prueba en conjunto con Raspberry Pi, que es un computador de placa única desarrollada en Reino Unido con el objetivo de estimular la enseñanza de ciencias de la computación. han permitido que este proyecto busque desarrollar la telemedicina en el sistema de salud nacional a partir de la incorporación de una plataforma portátil que permite realizar exámenes médicos, además almacenar los datos y poder enviarlos y visualizarlos vía Web, sin embargo, aún no se ha llegado al producto final.

Para llegar a un producto final que cumpla con las funciones y especificaciones mencionadas anteriormente, es necesario realizar un estudio minucioso y tener en consideración las características y funciones de cada uno de los componentes a utilizar. También se debe considerar y enfocar cada edición de este proyecto en el usuario final. Ya que debe ser un dispositivo fácil de transportar y utilizar. Para ello es necesario estudiar la compatibilidad de las conexiones, desarrollar un software manejable por cualquier persona y diseñar un esqueleto transportable y ergonómico.

Los protocolos de comunicación corresponden al conjunto de pautas que posibilitan a distintos elementos formar parte de un sistema, donde se establezcan comunicaciones entre sí y permitan el intercambio de información. Los microcontroladores poseen varios protocolos, siendo los más principales UART (Transmisor y Receptor Asíncrono Universal), USB (Bus serial universal), I2C (Circuito inter-integrado), SPI (Interfaz serial periférica).

Raspberry Pi como ordenador cuenta con todos los protocolos mencionados anteriormente y hace uso de ellos para realizar las lecturas a los distintos sensores o dispositivos que se le conectan, es importante que cuando se quiera implementar un sistema como éste, todos los dispositivos que se desean conectar ocupen protocolos de comunicación diferente, para que así la información no se tergiverse y se realicen lecturas erróneas. Y es aún más importante que los protocolos que utiliza, la plataforma E-Health sean leídos de forma inmediata sin ser interrumpidos por otras funciones que desee ejecutar el ordenador.

Con la idea de simplificar el uso de los ordenadores para usuarios de todo tipo y no sólo para los expertos, como lo sería en este caso la aplicación de este dispositivo médico en el sistema de salud. Se ha convertido en una práctica habitual utilizar metáforas visuales por medio de la llamada interfaz gráfica de usuario (IGU) para que el usuario interactúe y establezca un símbolo del sistema contacto más fácil e intuitivo con el ordenador.

Una interfaz es un medio que permite comunicar dos sistemas que no hablan el mismo lenguaje. Restringido a aspectos técnicos, se emplea el término interfaz para definir el juego de conexiones y dispositivos que hacen posible la comunicación entre dos sistemas. Sin embargo, cuando aquí se habla de interfaz se hace referencia a la cara visible de los programas tal y como se presenta a los usuarios para que interactúen con la máquina. La interfaz gráfica implica la presencia de un monitor de ordenador o pantalla constituida por una serie de menús e íconos que representan las opciones que el usuario puede tomar dentro del sistema.

Se requiere de un armazón que sea resistente, cómoda y ergonómica para el usuario. Para ello se realizó un diseño 3D, el cual corresponde a gráficos que se utilizan en una representación tridimensional de datos geométricos, que se pueden almacenar en un ordenador con el fin de conseguir una proyección visual en dos dimensiones, para ser mostrada en pantalla o impresa en un papel. Esto permite contar con una perspectiva diferente para los proyectos, ya que ayuda a plasmar y compartir con los demás las ideas que se tenga sobre la confección y organización de alguna pieza. Con ello verificar además la viabilidad de ver si lo que se imagina tiene una verdadera utilidad y aplicación.

Se cuenta con diferentes softwares, uno de ellos es SketchUp el cual permite modelado en 3D de edificios, paisajes, escenarios, mobiliario, personas y cualquier objeto o artículo que imagine el diseñador o dibujante, con el objetivo de que pueda usarse de una manera intuitiva y flexible.

Una impresora 3D corresponde a una máquina, que es capaz de transformar los diseños realizados en software como SketchUp y llevarlos a un plano tridimensional. Ya que, mediante el uso de materiales plásticos como PLA (ácido poli-láctico), se realiza un barrido de este material fundido sobre una plataforma sólida, permitiendo la obtención de la pieza diseñada en el software.

Este proyecto mostrará cómo se utiliza la plataforma E-Health, Raspberry Pi, una interfaz gráfica y el uso de diseños e impresiones 3D, que en conjunto permitirán visualizar, crear y confeccionar el dispositivo final médico que busca incorporar la telemedicina al sistema de salud.

### **Objetivo General**

- Mejorar las prestaciones de la plataforma E-Health para la toma de exámenes médicos, con el fin de obtener un dispositivo portátil final, que permita la aplicación de la telemedicina en el sistema de salud.

### **Objetivos Específicos**

- Revisión del estado del arte acerca de toma de Exámenes y del uso de la Plataforma E-Health para la toma de exámenes.
- Revisión de los proyectos de título realizados con la plataforma E-Health.
- Implementación de mejoras a la plataforma E-Health (Velocidades de respuesta, gráficas de los exámenes, código utilizado, entre otros).
- Mejorar la estructura y hardware del dispositivo propuesto.
- Llegar a un dispositivo portátil final que pueda competir con las nuevas incorporaciones al mercado, que están orientadas a los servicios de telemedicina.

# 1 Antecedentes del proyecto

A continuación, se presenta la base y fundamento de este proyecto, ya que se expone la problemática, luego se aborda el estado del arte y finalmente se propone la solución a la problemática planteada.

## 1.1 Problemática

El sistema de salud chileno hoy en día está siendo afectado por una carencia sistemática de profesionales médicos en el sistema público. A esto se suma la entrega de muy pocos recursos para el funcionamiento de los organismos de salud. Todo esto lleva a una mala administración y organización del sistema de salud y que se refleja principalmente en el continuo colapso y las largas horas de espera de los pacientes para ser atendidos en los hospitales y consultorios. [1]

Se debe considerar además que Chile es un país centralizado, por lo que la gran mayoría de especialistas se encuentra en Santiago, también cabe destacar que la geografía montañosa y segregada del territorio nacional, hace que existan distancias muy grandes entre los grandes hospitales y las comunidades rurales que sólo cuentan con consultorios, provocando que el acceso a áreas médicas especialistas sea más difícil.

Ahí surge la necesidad de implementar nuevas tecnologías que permitan resolver la mayor cantidad de carencias que presentan las personas de estas comunidades lejanas, permitiendo con ello mejorar el sistema de salud pública, por medio de la rapidez y eficiencia que se les puede proporcionar a partir de la incorporación de dichas tecnologías a los centros de salud más cercanos.

Este proyecto busca aportar la incorporación de la telemedicina a partir de un dispositivo portátil que permita realizar la toma de signos vitales y exámenes que comúnmente solo se pueden realizar en los grandes centros de salud, ya que podrá transmitir la información vía web y almacenarla en el dispositivo para que así pueda ser revisada a distancia por un especialista en tiempo real o posteriormente.

## 1.2 Estado del arte

### 1.2.1 Telemedicina

La telemedicina es la prestación de servicios de medicina a distancia, permite la interacción entre el paciente y el médico especialista. Para ello se emplean las tecnologías de la información y de las comunicaciones, para compartir los conocimientos médicos que van en busca de la mejora de salud del paciente de forma rápida y eficiente. Esto implica el uso de dispositivos electrónicos y protocolos de telecomunicaciones como herramienta de apoyo al diagnóstico clínico, como instrumento de capacitación y actualización médica, así como medio informativo para los pacientes.

Existen tres tipos de telemedicina:

- Monitorización de pacientes a distancia: Esto permite controlar en sus casas a los pacientes con enfermedades crónicas. Se consigue a través del uso de dispositivos que recopilan datos de nivel de azúcar en la sangre, presión arterial y otros signos vitales. Los cuidadores pueden revisar esos datos al momento desde el hospital, y decidir si es necesaria algún tipo de acción.
- Tecnología de almacenamiento y envío: Consiste en el almacenamiento de datos clínicos para su envío a otros centros médicos. Por ejemplo, así se pueden tomar imágenes de rayos X en el ambulatorio y enviarlas al momento a un centro especializado para su interpretación.
- Telemedicina interactiva: Este tipo de telemedicina permite a médicos y pacientes comunicarse en tiempo real. Básicamente se trata de una videoconferencia, para la cual el paciente puede quedarse en su domicilio o acudir al centro de salud más cercano. [2]

Un sistema de telemedicina opera a partir de la solicitud de un centro de salud que presenta carencia de especialistas. Éste será asistido por uno de mayor envergadura y que cuente con el personal y tiempo adecuado para atender a los pacientes de forma remota, gracias a esto las personas podrán atender sus necesidades en el centro más cercano, permitiéndoles el ahorro de tiempo y dinero, lo que lleva una mejora importante en el sistema de salud.

A continuación, se presenta en la Figura 1-1 un sistema de telemedicina.

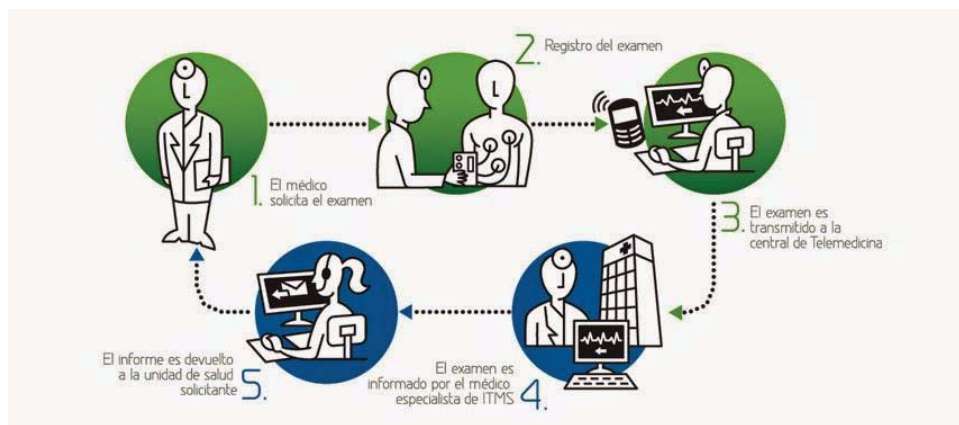


Figura 1-1 Sistema de telemedicina [3]

Para que un sistema de estas características funcione adecuadamente, se debe contar con:

- Equipos capaces de comunicarse a través de redes de telecomunicación.
- Medios de comunicación (Satelital, Radio Frecuencia, Internet)
- Protocolos y estándares de comunicación.
- Centros de salud dedicados que presten los servicios médicos requeridos. Para ello, deben contar con la infraestructura, tiempo y especialistas adecuados para la labor.

El uso de este nuevo tipo de servicio médico permite obtener una serie de beneficios:

- Optimización en la atención integral del paciente y la asignación de recursos médicos.
- Reducción en los costos y tiempos requeridos por las personas para optar a la atención de un especialista.
- Disminución del colapso de los grandes centros de salud, mejorando los tiempos de respuesta en las interconsultas.
- Contribuye a la formación y actualización académica del personal utilizando herramientas de educación virtual y teleconferencia.
- Fomenta la equidad llevando la atención a las áreas aisladas.

### 1.2.2 Telemedicina a nivel mundial y nacional

En la actualidad la telemedicina sigue ganando terreno tanto en los países desarrollados como en los que están en vías de desarrollo. Los usos y necesidades de telesalud varían entre el mundo desarrollado y en desarrollo; ya que los últimos luchan con enfermedades transmisibles y no transmisibles, y con muy pocos recursos. Por eso, los expertos coinciden en que la telemedicina mejorará la atención clínica en esas zonas. En los países desarrollados la telemedicina puede dar más poder a los profesionales sanitarios locales, mediante un apoyo médico a distancia, mejorando considerablemente la autosuficiencia. Existen muchos proyectos de telesalud en toda América Latina y, Asia y África, pero aún con poca evidencia publicada.

Sin embargo, aunque las evidencias demuestran que la telemedicina se ha utilizado en prácticamente todos los países del mundo, son pocos en los que esté completamente integrada en la práctica médica.

Los países que actualmente cuentan con los sistemas de telemedicina más avanzados son:

- Estados Unidos donde hoy la gama de aplicaciones en todo el país, a nivel de hospitales y facultades de medicina, es amplia. Se emplean sistemas satelitales y de fibra óptica entre los diferentes centros médicos, y tecnología inalámbrica doméstica para la atención domiciliaria. Entre los factores que han incidido en la utilización de esta tecnología están el alto índice de subespecialidades médicas y la elevada densidad demográfica de su territorio. La crisis que atraviesa el sector salud ha propiciado un aumento exponencial en su uso. Al mismo tiempo, los altos precios en la atención médica han obligado al sistema general de salud a aplicar tecnologías de comunicación en medicina, para reducir en parte los costos generados a la

población. Las inversiones en el sector privado y público, en telemedicina, ascienden a 1,4 mil millones de dólares al año.

- Reino Unido, uno de los factores que más han influido en el desarrollo de estos sistemas es el alto costo de la atención en salud para sus habitantes. También incide el elevado porcentaje de población rural productiva, la cual es atendida en hospitales de baja complejidad. Sus fortalezas en telemedicina incluyen telecardiología (análisis de pulso y realización de ECG), teleradiología (transmisión de imágenes de radiografía convencional, tomografía computarizada, resonancia magnética, ultrasonido y mamografías), teledermatología (transmisión de imágenes de lesiones dermatológicas, incluyendo datos y texto en tiempo real), telenfermería (análisis y diagnóstico de patologías comunes en tiempo real). [4] [5]

Los médicos rara vez realizan procedimientos médicos de forma remota a través de la telemedicina; en cambio, la telemedicina se utiliza principalmente como una herramienta de diagnóstico para apoyar el sistema de salud local.

En Chile cada día se desarrollan más las aplicaciones en telemedicina. Hoy en día el ministerio de salud trabaja con un sistema de teleconsulta y teleimagenología que permite evaluar exámenes y otorgar atención especializada al paciente. Por medio de la teleimagenología se transmiten imágenes digitales de las tomografías, electrocardiogramas y mamografías, para la generación de informes diagnósticos en 24 horas. Esto ha sido de vital importancia y ayuda a los pacientes de las regiones más remotas, que pueden así ser evaluados por médicos especialistas de Santiago. También se incluyen los pacientes con enfermedades crónicas. La incorporación de estos pacientes a un programa de telemonitoreo, permite reducir la tendencia a la complicación de la enfermedad, ellos cuentan con servicios como Accuhealth, que ofrece monitoreo a distancia para pacientes del sector público y privado. Esta empresa desarrolló un modelo tecnológico centrado en el control y monitoreo de patologías como diabetes mellitus tipo 1 o 2, hipertensión arterial, asma e insuficiencia cardiaca. [6]

El 27 de mayo de 2016 el consejo universitario de la Universidad de Chile aprobó la creación del Centro de Informática Médica y Telemedicina (CIMT), posteriormente dicha universidad empezó el proyecto ECHO (Extensión para los resultados de atención médica en la comunidad), el cual tiene como objetivos conectar a equipos de salud de zonas alejadas con especialistas universitarios de grandes centros médicos a fin de mejorar la atención de los pacientes, democratizar el conocimiento y el acceso a la salud. Actualmente dicha universidad se posiciona en redes internacionales de telemedicina. [7]

### 1.2.3 Equipos actuales para telemedicina

Existen empresas que están orientadas a la venta de equipos para telemedicina, las cuales incluyen una amplia gama de productos, que son de excelente calidad y poseen una plataforma muy cómoda y fácil de utilizar, pero el inconveniente se presenta en el precio de cada uno de estos dispositivos, ya que es muy elevado. Se puede apreciar en la Figura 1-2 las empresas que distribuyen productos orientados a este tipo de tecnología.





Figura 1-2 Empresas con productos para telemedicina

En la Figura 1-3 se puede apreciar un dispositivo portátil que realiza ECG y además cuenta con Bluetooth para poder conectarse desde el celular, este dispositivo es de la empresa CardioVex y tiene un precio de USD\$ 23.542, lo que se traduce en \$15.297.191, siendo muy costoso para poder incorporarlo dentro de un sistema de telemedicina, lo que demuestra que un sistema mucho más completo, tendrá un costo más elevado.



Figura 1-3 ECG portátil con Bluetooth [8]

En marzo del presente año la empresa Libelium que se especializa en la creación de plataformas electrónicas lanzó un dispositivo portátil orientado a la telemedicina, este dispositivo se denomina MySignal e incorpora un kit completo (Figura 1-4) que incluye más de 15 sensores que pueden ser utilizados para supervisar señales biométricas. Todos los datos recogidos por MySignals son encriptados y enviados a la cuenta privada del usuario en la nube Libelium a través de Wi-Fi o Bluetooth. Los datos pueden ser visualizados en una tableta o teléfono inteligente con Android o iPhone Apps. Este dispositivo tiene un costo de €1.710, los cuales son unos \$1.200.000,

lo que si corresponde a un precio bastante accesible para los centros de salud menores y que cuentan con menos recursos, este representa una alternativa ideal para poner en marcha un sistema de telemedicina en todo el país.



Figura 1-4 Kit MySignal SW BLE [9]

### 1.3 Solución propuesta

Este proyecto que ya cuenta con dos iteraciones plantea como solución la creación de un dispositivo portátil para realizar la toma de exámenes médicos, la idea principal es hacer una interfaz gráfica que muestre los signos vitales a un médico, paramédico o enfermero en tiempo real, también permite almacenar los datos y enviarlos a un servidor, con el fin de comunicar la información médica de un punto a otro, cumpliendo con los principios básicos de la telemedicina.

#### 1.3.1 Plataforma programable

Raspberry Pi es un ordenador de placa reducida y bajo costo, desarrollado en el Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas. Este dispositivo soporta varios componentes y puede realizar diversas funciones similares a un computador personal normal, como procesamiento de textos y datos, reproducción de videos en alta definición y conexión a Internet vía Ethernet o Wi-Fi.

Existen varios modelos, que poseen características diferentes ya que están orientados a distintos usos, en este caso se utilizó la Raspberry Pi 2 Model B (Figura 1-5).

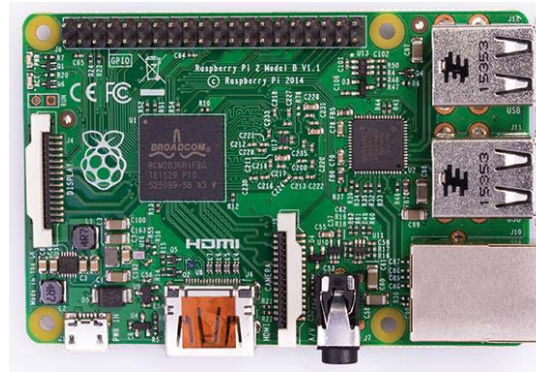


Figura 1-5 Raspberry Pi 2 Model B [10]

La plataforma presenta las siguientes características:

- Procesador Broadcom BCM2836 de 900 MHz ARM Cortex-A7 de cuatro núcleos con GPU (Unidad de procesamiento gráfico) VideoCore IV de doble núcleo.
- GPU proporciona una tecnología Open GL ES 2.0, hardware acelerado OpenVG y admite imágenes de alta resolución 1080p30 H.264.
- GPU tiene una capacidad de 1 Gpixel/s, 1,5 Gtexel/s o 24 GFLOPs con filtrado e infraestructura DMA (Acceso directo a memoria).
- SDRAM LPDDR2 de 1 GB.
- Dimensiones: 86 x 56 x 20 mm.

Su funcionamiento se basa en la ejecución de un sistema operativo que se instala en la tarjeta micro SD, generalmente se utiliza Linux, dentro de ese sistema operativo se debe programar para habilitar los buses y GPIO (Entrada/Salida de Propósito General), además se debe alimentar con +5 [V] a 2[A] a través del conector hembra microUSB.

### 1.3.2 Plataforma médica

La plataforma E-Health (Figura 1-6) ha sido diseñada Cooking Hacks con el objetivo de ir en ayuda de investigadores, desarrolladores y artistas para medir datos de sensores biométricos con fines de experimentación y de prueba.

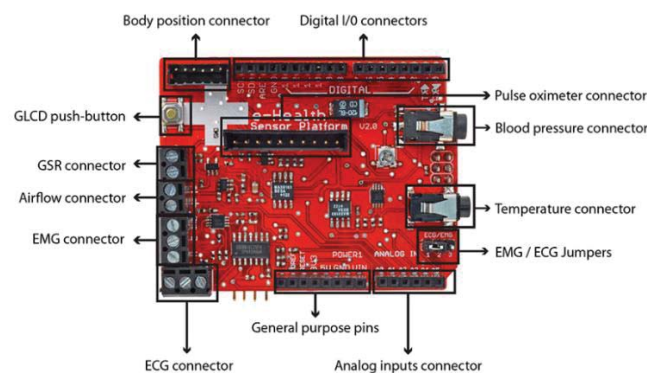


Figura 1-6 Plataforma E-Health V2.0 [11]

Toda la información obtenida por los sensores y plataforma, se puede visualizar en tiempo real. No existe aplicación o interfaz gráfica para Windows, Linux o OSX. Pero si existe una aplicación para celular donde se puede ver toda la información. Para ver los datos obtenidos por los sensores, a través de la consola de la Raspberry pi se aplican comandos ya definidos por la plataforma, como ya se mencionó antes, E-Health ha sido desarrollado para usuarios de Arduino y Raspberry Pi.

Toda la información biométrica reunida se puede enviar de forma inalámbrica utilizando distintos tipos de conectividad las cuales son:

- Wi-Fi
- 3G (Tercera generación de transmisión de voz y datos a través de telefonía móvil) y GPRS (servicio general de paquetes vía radio)
- Bluetooth
- ZigBee 802.15.4

Hay que aclarar que estos protocolos de comunicación inalámbricas no vienen incluidas en el kit, y tampoco en la Raspberry Pi 2 que se ocupa para el proyecto.

Hay 9 sensores incluidos en el kit (Figura 1-67), los cuales son:

- Detección de la posición del cuerpo.
- Sensor de glucosa.
- Sensor de temperatura.
- Sensor de presión arterial.
- Sensor de pulso y oxígeno.
- Sensor de flujo de aire (respiración).
- Mediciones de la respuesta galvánica de la piel.
- Sensor de electrocardiograma.
- Sensor de electromiografía.



Figura 1-7 Sensores E-Health [11]

### 1.3.3 Ensamblaje de Plataforma E-Health con Raspberry Pi

En este proceso es necesario un adaptador llamado puente de conexión Raspberry Pi (Figura 1-8 y Figura 1-9), el cual fue diseñado para funcionar como un vínculo entre Raspberry Pi, es decir, permite la utilización de cualquier módulo, tarjeta o escudo diseñado para una Raspberry Pi, además con este puente también se puede conectar cualquier sensor analógico o digital utilizando el PinOut de Raspberry Pi aprovechando al máximo sus capacidades.



Figura 1-8 Conexión E-Health a Raspberry Pi [11]



Figura 1-9 Plataforma E-Health conectada a Raspberry Pi [11]

### 1.3.4 Sensores y comandos para obtención de datos en consola

Como se mencionó anteriormente que la plataforma cuenta con una gran variedad de sensores, los cuales cuentan con una conexión específica, permitiendo que su uso sea mucho más rápido y fácil. También posee las librerías y códigos necesarios, para realizar una lectura simple y precisa. Estas librerías se pueden descargar en la página oficial de Cooking Hacks.

Las librerías que pertenecen a la carpeta ArduPi y que sirve para ver los datos son los siguientes

- eHealth.h: es la librería oficial para operar con la placa de sensores

- arduPi.h: Es una librería de C++ que permite escribir programas para Raspberry Pi como si estuviera escribiendo un programa de Arduino, es decir, esta librería permite trabajar con la placa de conversión de Arduino a Raspberry Pi 2
- time.h: se utiliza para obtener el tiempo de ejecución de cada código o programa

### Sensor de temperatura

La conexión del sensor en la plataforma se hace directamente en el conector. y en el paciente se puede hacer en la boca, la axila, la piel y la oreja, tal como se muestra en la Figura 1-10.



Figura 1-10 Lectura sensor de temperatura [11]

El comando para obtener los datos es: `$ sudo ./temperatura`

Después de enviar el comando para obtener los datos en consola, el sensor tarda en enviar 2 minutos la temperatura corporal. El motivo principal es para que tenga un tiempo determinado para enviar un correcto dato de temperatura.

### Pulso y oxígeno en la sangre

Su conexión con la plataforma es muy sencilla, se conecta directamente en la ranura del oxímetro y en el paciente el sensor se conecta comúnmente en el dedo índice de éste como se muestra en la Figura 1-11.

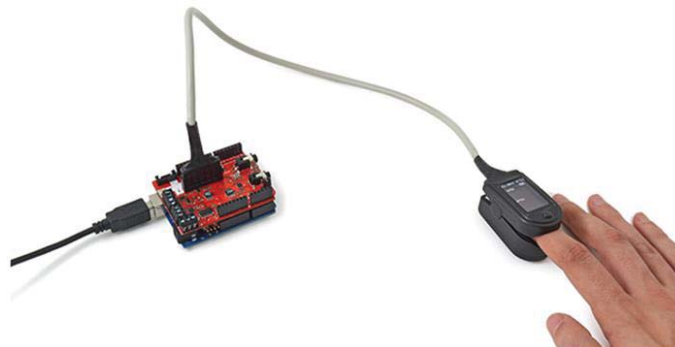


Figura 1-11 Sensor pulso y oxígeno. [11]

El comando para obtener los datos es: `$ sudo ./saturacion`

Cuando se envía el comando para obtener datos, éste empieza a leer los datos recibidos por el sensor. Éste muestra las pulsaciones y el oxígeno que hay en nuestro cuerpo, se demora aproximadamente 30 segundos.

### Sensor de flujo de aire

La conexión de este sensor en la plataforma debe ser con un cuidado especial, ya que se debe tener en cuenta la polaridad de los conectores (Figura 1-12) debido a que si se conecta de forma errónea los datos obtenidos no serán los correctos. La conexión con el paciente es muy simple, se coloca el sensor en las fosas nasales y se fijan por detrás de las orejas.

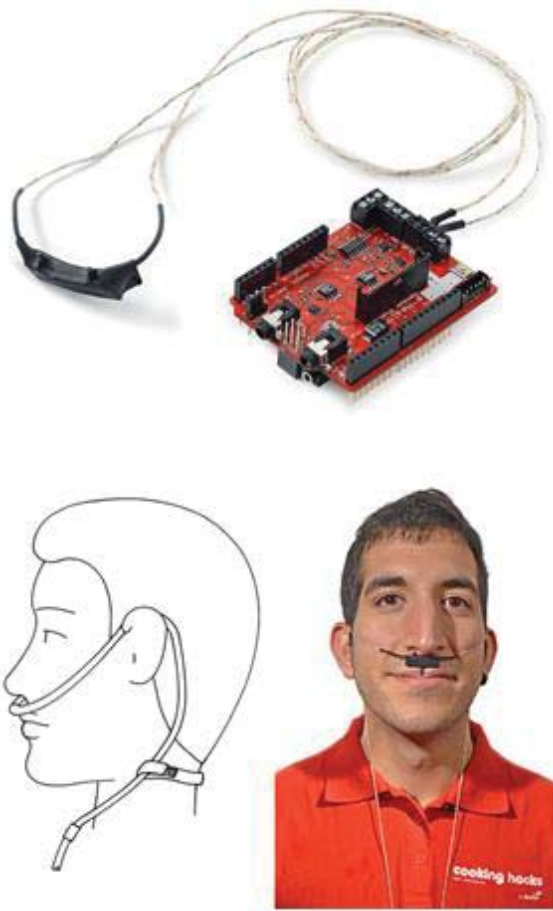


Figura 1-12 Sensor de flujo de aire [11]

El comando para obtener los datos es: `$ sudo ./respiracion`

Cuando se envía el comando por consola para obtener datos, se ve como varía la respiración respecto al tiempo que pasa, esto es una obtención de datos en tiempo real y su duración es de 1 minuto. Éste sensor utiliza la librería de tiempo.

### Presión Arterial

Para poder conectar el sensor a la plataforma, primero se necesita al menos una medida en la memoria del sensor de presión arterial. Para esto, se coloca el sensor llamado esfigmomanómetro en la zona del bíceps tal como se muestra en la Figura 1-13.

Luego con el fin de extraer los datos del sensor, se conecta un cable adaptador al terminal Jack de la plataforma y el otro terminal miniUSB al esfigmomanómetro, como en la Figura 1-14.



Figura 1-13 Sensor de presión arterial [11]



Figura 1-14 Extracción de datos del sensor [11]

El comando para obtener los datos es: `$ sudo ./Presion`

Cuando se envía el comando para obtener datos, el sensor tiene una memoria interna la cual guarda la información que ha tomado, más la que está tomando en el momento, lo que se muestra en pantalla es el historial de medidas que ha hecho el sensor.



## Glucómetro

Antes de conectar el glucómetro a la plataforma, se necesita al menos una medida en la memoria (Figura 1-15). Después se puede obtener toda la información contenida en la memoria del glucómetro (fecha, valor de glucosa) conectándolo tal como se muestra en la Figura 1-16.



Figura 1-15 Sensor glucómetro [11]



Figura 1-16 Conexión glucómetro [11]

El comando para obtener los datos es: `$ sudo ./glucometro`

Al enviar el comando para obtener datos, este dispositivo tiene una memoria interna que guarda los datos que ya ha tomado más el que está en ejecución. Por lo tanto, cuando se activa el sensor, en pantalla se muestra el historial de mediciones que ha hecho.

## Electrocardiograma (ECG)

Para la conexión de este sensor en el paciente, primero se debe elegir qué derivación precordial se usará. Esta plataforma sólo admite la medición en tres derivaciones, debido a que sólo posee tres electrodos, por lo tanto se usará la siguiente configuración (Figura 1-17), con esto se logra ver el cambio de frecuencia cardíaca, que es el objetivo de este examen. Es importante destacar que

este examen solo se puede realizar si el paciente esta acostado sobre una camilla. De lo contrario los resultados obtenidos no serían óptimos.

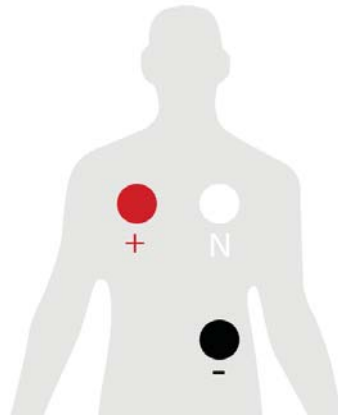


Figura 1-17 Derivaciones ECG [11]

Luego para conectar los electrodos a la plataforma, primero se debe seleccionar el interruptor en modo ECG y luego se conectan teniendo en cuenta la polaridad de cada uno tal como se muestra en la Figura 1-18.



Figura 1-18 Conexión sensor ECG [11]

El comando para obtener los datos es: `$ sudo ./ecg`

Cuando se envía el comando para obtener datos, Este código realiza la misma tarea que la del sensor de flujo de aire con la diferencia de que en vez de recibir el valor análogo del sensor la función de éste realiza una conversión y devuelve el valor en mV.

### **Electromiograma (EMG)**

Al igual que en el ECG primero se debe elegir el músculo a medir, con el objetivo de poder medir el cambio de intensidad del músculo, en este proyecto la mayoría de las pruebas se harán en el músculo bíceps o tríceps tal como se muestra en la Figura 1-19.

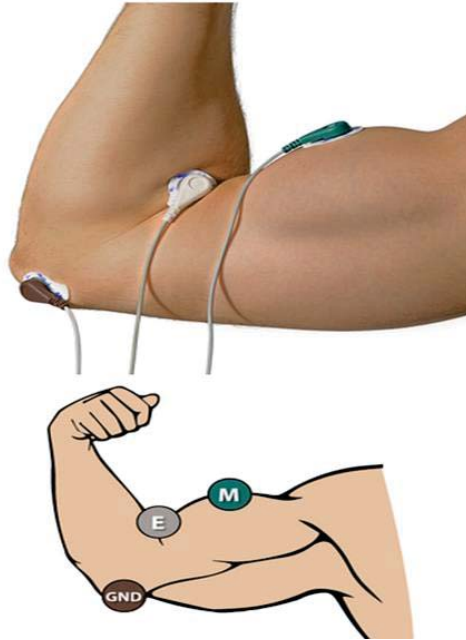


Figura 1-19 Derivaciones EMG [11]

Para obtener los valores medidos se conectan los electrodos (Figura 1-20) teniendo en cuenta el orden en el cual se pusieron en el músculo elegido anteriormente, también está la opción de regular la ganancia de la señal para que esta sea más sensible al cambio de la intensidad del músculo, pero para dejarlo universal se optó por dejar la misma ganancia para todas las mediciones.

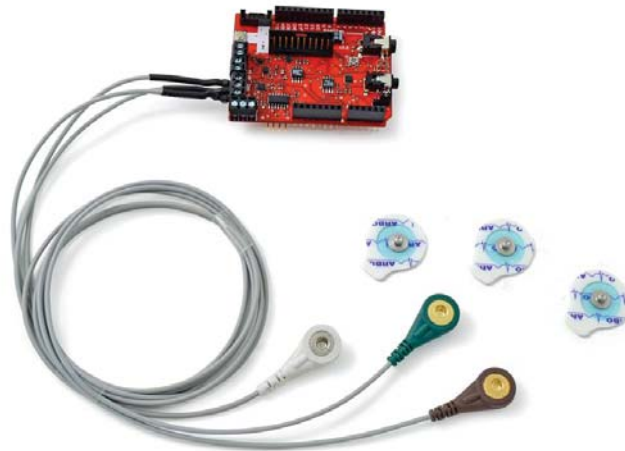


Figura 1-20 Conexión electrodos [11]

El comando para obtener los datos es: `$ sudo ./EMG`

Cuando se envía el comando para obtener datos, el código fuente del sensor permite ir recibiendo los valores análogos entregados por el sensor al mover algún musculo del cuerpo.

## 1.4 Estado de avance

Como este proyecto cuenta con antecedentes de avance, en esta sección se analizará lo realizado en cada una de ellos. Resumiendo, por separado la conformación del hardware de la plataforma y la edición en el software que se constituye principalmente de una interfaz gráfica para el sistema.

### 1.4.1 Software del sistema

En la primera iteración se activó cada sensor de la plataforma E-Health a través de las librerías que se mencionaron anteriormente, cada uno de ellos se activa mediante una función, las cuales están insertas en el código de la librería eHeath.h, para que la Raspberry Pi pueda procesar estas funciones se debe utilizar la librería ArduPi, que contiene los ficheros arduPi.cpp, arduPi.h y arduPi\_template.cpp, también se utiliza la librería time.h para contabilizar el tiempo de duración del examen. Cada una de las funciones y códigos de programación se encuentran modificados con el fin de poder obtener la mayor exactitud posible en la toma de cada examen.

En la segunda iteración se creó una interfaz gráfica que permite visualizar los resultados de los exámenes en tiempo real y a través de Internet. Para ello se instaló LAMP; Linux, APACHE (servidor web), MySQL (gestor de base de datos), PHP (procesador de hipertexto) que corresponde a un conjunto de componentes que por medio de la Raspberry Pi permite crear el servidor web para la plataforma E-Health. En la Figura 1-21 se ve el procedimiento de un servidor web LAMP.



Figura 1-21 Red LAMP [12].

Para almacenar los datos obtenidos por la plataforma E-Health en una carpeta que se muestre en el servidor Apache, se debe añadir la dirección /var/www. En esa carpeta, se copiará toda la información, o sea se moverá todo el contenido de la carpeta ArduPi donde se almacenan las librerías y programas para hacer funcionar los sensores. Esto se hace para que el servidor Apache pueda reconocer las librerías y programas que contiene dicha carpeta.


Para hacer el programa, hay que crear un archivo HTML (lenguaje de marcas de hipertexto) con extensión .php, en este caso será index.php donde se almacena el programa principal para que el usuario pueda presionar un botón y que obtenga los resultados de la medición por pantalla.

El programa obtiene los valores de forma discreta, por lo tanto es necesario añadir la librería para representar los datos obtenidos en los exámenes ECG, EMG y flujo de aire, mediante una gráfica ya que estos requieren de un análisis diferente.

Para ello se utilizó la librería jgraph, que se añade dentro de la carpeta www donde se encuentran todos los códigos y programas a ocupar. Luego se crearon tres archivos nuevos, llamados, crearemng.php, crearecg.php y crearresp.php, con los cuales se almacena el código necesario para obtener los datos y graficarlos al término de la medición.

La plataforma permite obtener los datos finales en PDF a través de la herramienta denominada Fpdf, que permite desarrollar aplicaciones en línea con el lenguaje PHP. La finalidad de crear un PDF en la plataforma, es para que obtenga los datos finales de los sensores y pueda crear un reporte completo del paciente, esto permite que se puedan mostrar los datos obtenidos en cada página de este PDF, y comparar los datos de los sensores con los rangos aceptados como buenas mediciones. Al igual que la librería Jpgraph se tiene que descargar la librería FPDF y moverla a la carpeta donde se está desarrollando nuestra aplicación, que es en la carpeta /var/www.

En las siguientes imágenes se muestran todas las ventanas que posee la interfaz gráfica y se explica brevemente cuál es su respectiva función. En la Figura 1-22 se muestra el interfaz de bienvenida y con ello una pequeña restricción en la utilización de la plataforma. Para poder acceder a la siguiente ventana se debe ocupar el nombre de usuario “user1” y la contraseña “pass1”.



**Teclea un Usuario y contraseña**

Usuario:

Contraseña:

**Entrar**

Figura 1-22 Interfaz inicial.

Una vez escrita la información, se despliega la ventana en donde se deben ingresar los datos del paciente (Figura 1-23), con la finalidad de utilizar automáticamente esos datos en el reporte médico generado.

Figura 1-23 Datos del paciente.

Luego de que todos los datos estén ingresados correctamente, se despliega el interfaz de Menú Principal (Figura 1-24), es acá donde se mostrarán los datos de las diferentes mediciones realizadas por los sensores. Además, se ve la sección de los exámenes que requieren una interpretación gráfica como el flujo respiratorio, electrocardiograma y electromiograma. En la parte derecha se despliegan opciones como observaciones, reporte médico, borrar mediciones y cierre de sesión que facilitan el uso de esta plataforma.

Figura 1-24 Menú Principal.

Las mediciones que se representan gráficamente son tomadas en otra ventana, que se despliega una vez que se presiona el botón Gráficas (ver Figura 1-25), ahí al apretar los botones Respiración, Electrocardiograma y Electromiograma, aparece una nueva ventana con cada examen correspondiente.



Figura 1-25 Ventana Gráficas.

Una vez que se obtienen todos los parámetros, se puede volver al menú principal y seleccionar observaciones (ver Figura 1-26). Ésta permite escribir cualquier anomalía importante que presente el paciente.

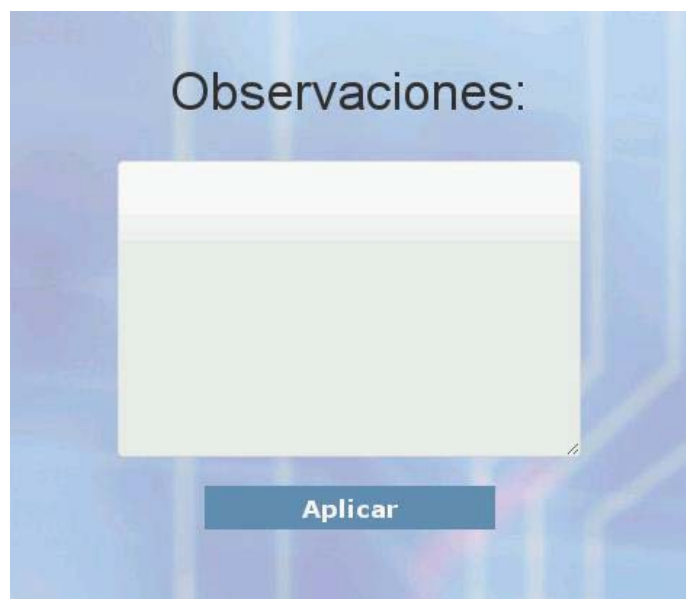


Figura 1-26 Ventana observaciones.

Si se selecciona reporte médico, se puede acceder a la parte final del proceso, donde al elegir la opción Crear Reporte mostrada en la Figura 1-27, se realizará un documento PDF que contiene todas las mediciones y los datos del paciente. Luego de crear el reporte, estos datos se pueden enviar vía mail relleno las casillas destinatario, asunto y mensaje.



**Reporte Medico**

**Crear Reporte**

Para:

Asunto:

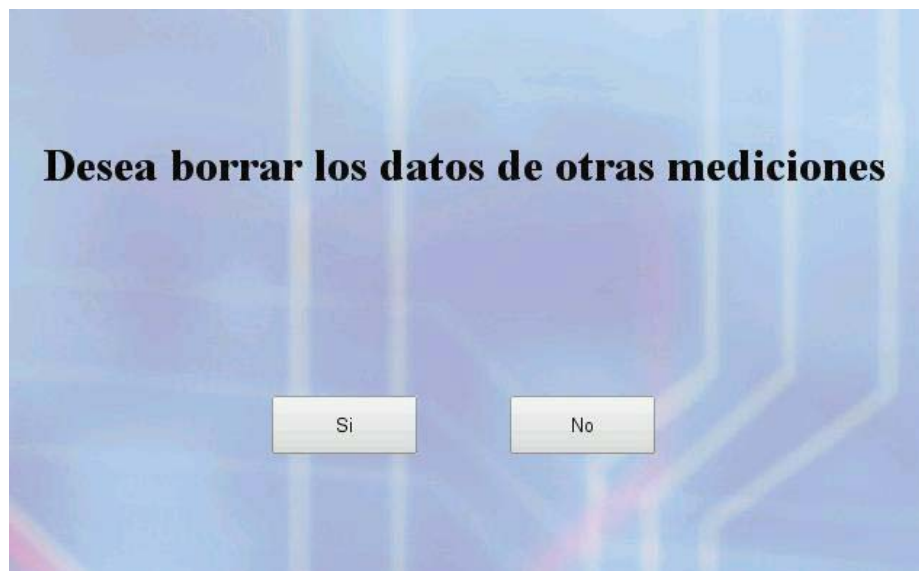
Mensaje:

**Enviar mail**

**Volver**

Figura 1-27 Crear reporte médico.

Posteriormente, si se elige Borrar mediciones, se despliega la ventana mostrada en la Figura 1-28, la cual elimina todos los datos tomados por el médico. Finalmente, al presionar cerrar sesión se vuelve a la interfaz que permite el acceso solo con el registro de usuario.



**Desea borrar los datos de otras mediciones**

Si No

Figura 1-28 Eliminación de mediciones.



Si bien esta interfaz cumple su funcionalidad que es la obtención de los datos médicos, tiene serios inconvenientes que impiden un buen funcionamiento de ella. Al estar creada por diferentes subprogramas, alrededor de 70, el pasar de una ventana a otra implica mucho tiempo. Tampoco posee un estilo genérico que se solicite cuando se requiera su utilización, si no que se carga en cada uno de los programas principales. Por eso uno de los principales cambios es mejorar la rapidez y funcionalidad de este software, además del aspecto visual.

### 1.4.2 Hardware del sistema

En la primera iteración la plataforma contaba con el ensamblaje que se muestra en la sección 1.3.3 y una caja de PLA (Figura 1-29) que se imprimió en la impresora Makerbot Replicator 2 en el Laboratorio de robótica, para realizar los exámenes se realizaban las conexiones de los sensores por los orificios que posee la caja y se conectaba directamente la plataforma a un Notebook.



Figura 1-29 Caja plataforma primera iteración

En la segunda iteración se incorporó una pantalla táctil a la plataforma para así poder tener mayor portabilidad y comodidad, ya que al tener una pantalla táctil, se puede ingresar u obtener datos sin utilizar un Notebook, Además de la pantalla, se integró lo que es un módulo de internet 3G para que la plataforma pueda ser usada en cualquier parte de Chile que tenga cobertura 3G. Además una batería recargable más una carcasa para el equipo completo.

La pantalla viene con una tarjeta de adaptación entre pantalla y Raspberry, el cable conector DSI (interfaz serial de monitor), 4 separadores, 4 tornillos y 4 cables hembra-hembra, el Pack de la pantalla se ve en la Figura 1-30. También dispone de los orificios listos para poder encajar de buena forma la pantalla con la Raspberry Pi como se ve en la Figura 1-31.



Figura 1-30 Pantalla táctil para Raspberry Pi [13]



Figura 1-31 Conexión pantalla táctil a Raspberry Pi [13]

La imagen se transmite directamente a la pantalla y se puede visualizar la interfaz de la Raspberry Pi como aparece en la Figura 1-32.



Figura 1-32 Interfaz Raspberry Pi

La finalidad de añadir un módulo 3G a la plataforma es que sea aún más portátil, ya que con el módulo no va a depender de una conexión WIFI o de Ethernet, con esto se puede llegar a lugares impensados, ya que casi la totalidad del territorio chileno está cubierto por señal 3G. Para ello se utiliza un Dongle Huawei E303, el cual aparece en la Figura 1-33.



Figura 1-33 Dongle Huawei E303 [14]

Sin una carcasa para almacenar lo que es la pantalla, la plataforma, batería y módulo 3G, no sería efectivo para comercializar este producto. Por eso se dedicó al diseño de la carcasa en el software Autodesk Inventor, la que se muestra en la Figura 1-34.

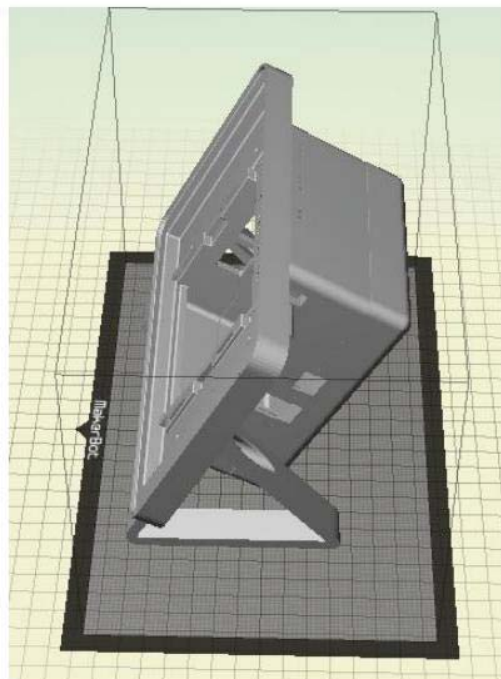


Figura 1-34 Carcasa diseñada en Inventor

## 1 Antecedentes del proyecto

Actualmente la plataforma se encuentra tal como se muestra en la Figura 1-35 ya que la pantalla no era propia del proyecto, además la carcasa 3D no se encuentra en su totalidad, los sensores están en buenas condiciones, pero así como está el dispositivo claramente no es para nada comercializable.

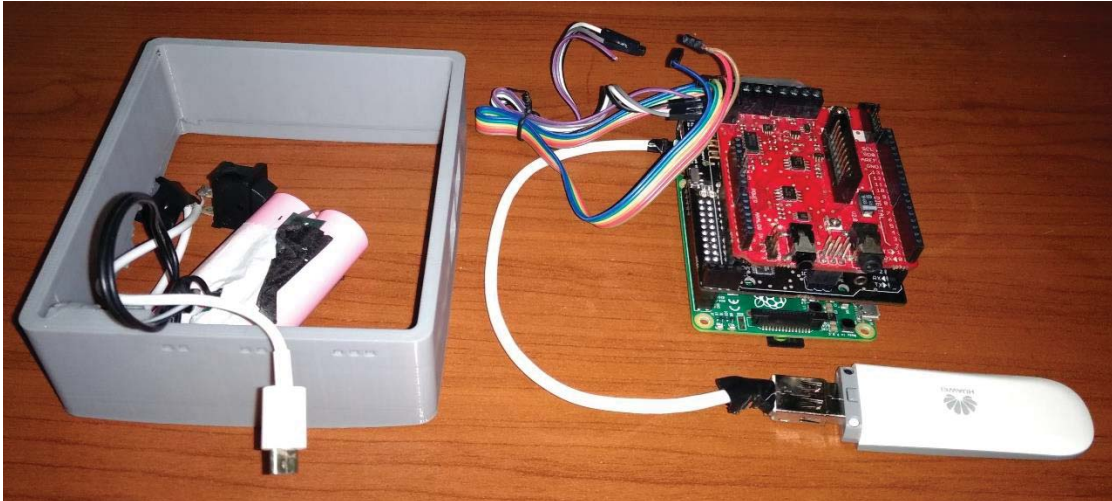


Figura 1-35 Estado actual de la plataforma

Finalmente al realizar la conexión de todos los sensores y reunir todas las piezas que conforman la base del dispositivo portátil, el dispositivo luce tal cual se muestra en la Figura 1-36.



Figura 1-36 Plataforma con sensores conectados.

## 2 Consideraciones técnicas para mejorar el dispositivo

Antes de incorporar a la plataforma E-Health los nuevos dispositivos, se debe hacer un estudio completo de los protocolos que rigen la comunicación de los microcontroladores, ya que en este caso cada dispositivo posee al menos uno de ellos para realizar sus funciones, en especial la Raspberry que es el punto de convergencia de todos los dispositivos que conforman el sistema.

### 2.1 Protocolos de comunicación

Corresponden a un conjunto de reglas y normas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellos para transmitir información por medio de cualquier tipo de variación de una magnitud física.

#### 2.1.1 UART (Transmisor y Receptor Asíncrono Universal)

Es un protocolo de comunicación en serie con dos líneas de señal de datos por medio de cables, con los que se crean las etiquetas Rx y Tx, dado esta comunicación en serie se utiliza comúnmente para transmitir y recibir la señal. UART toma bits de datos y los envía individualmente de manera secuencial por medio de modo halfduplex, donde no se puede enviar y recibir la información al mismo tiempo, la mayoría de los controladores de hardware cuentan con este protocolo de comunicación y utilizan una sola línea de datos para transmitir y recibir. Maneja 8 bits de datos, con un bit de inicio y otro de termino, donde se transmiten desde el bit más significativo al menos significativo, se da prueba de esto en la Figura 2-1. [15]

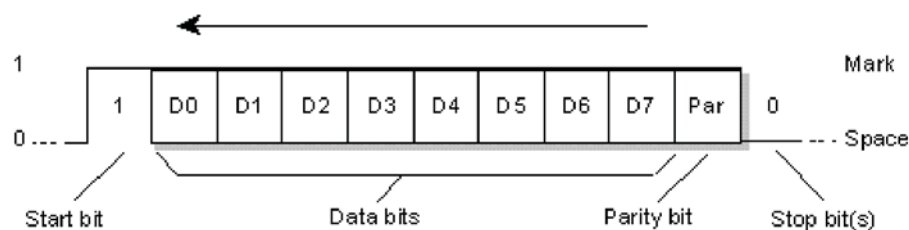


Figura 2-1 Protocolo UART. [15]

### 2.1.2 USB (Bus serial universal)

Protocolo de comunicación en serie con dos líneas de señal de datos por medio de cables, en donde las líneas de señal del cable de datos se denominan D+ y D-, se utiliza para comunicarse con los periféricos del sistema. La comunicación USB requiere un software de controlador basado en la funcionalidad del sistema y el USB conectado puede transferir datos en el bus sin ninguna petición del computador a mando. La transferencia de datos se puede realizar por medio de diversos modos, el primero de ellos es el modo lento de la velocidad 10kbps a 100 kbps, el segundo es el modo de velocidad completa de 500kbps a 10mbps, y el último modo es de alta velocidad 25mbps a 400mbps. La longitud máxima del cable USB es de 4 metros. En la Figura 2-2 se muestra un ejemplo de la secuencia utilizada para la comunicación en este protocolo. [16]

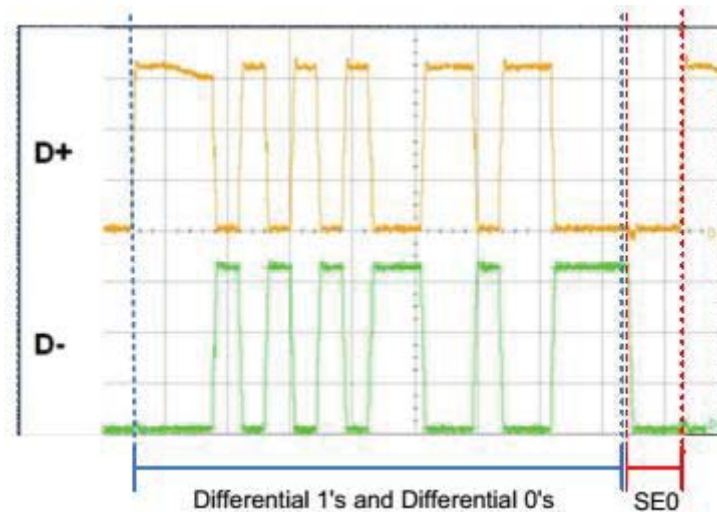


Figura 2-2 Protocolo USB. [16]

### 2.1.3 I2C (Circuito inter-integrado)

Este protocolo requiere de dos cables que conectan todos los periféricos al microcontrolador, uno de ellos es el SDA (línea de datos serie) y el otro el SCL (línea de reloj serie) para transportar información entre dispositivos. Su modo de comunicación se hace de maestro a esclavo, donde cada esclavo tiene una dirección única, el dispositivo maestro envía a la dirección del dispositivo esclavo el indicador de lectura/escritura, cuando la dirección coincide con cualquier dispositivo esclavo que esté activado, los dispositivos esclavos restantes se llevan a modo de desactivación y la comunicación se realiza, es así como se lleva a cabo la transmisión y recepción de los datos (Figura 2-3). El transmisor envía datos de 8 bits, el receptor responde con 1 bit de confirmación, cuando se completa la comunicación, el maestro emite el bit de detención, que finaliza la transferencia de información. I2C fue desarrollado por Philips Semiconductors, cuyo propósito original era proporcionar una manera fácil de conectar la CPU (unidad central de procesamiento) a chips periféricos, ya que los dispositivos periféricos en los sistemas embebidos a menudo están conectados al microcontrolador como dispositivos con asignación de memoria. [17]

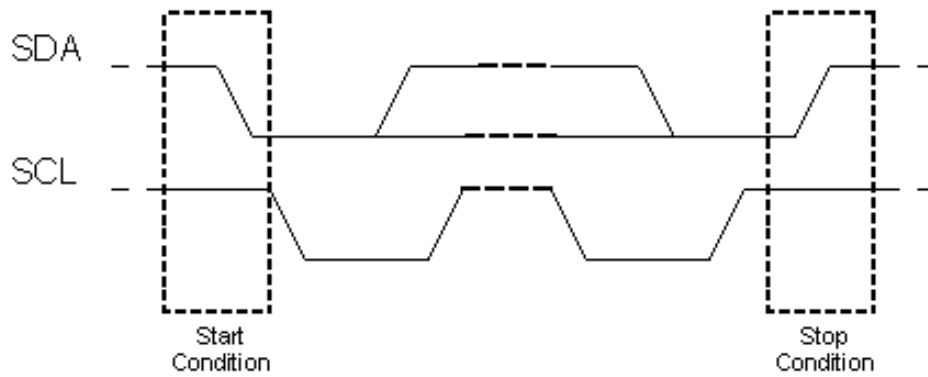


Figura 2-3 Protocolo I2C. [17]

### 2.1.4 SPI (Interfaz serial periférica)

Protocolo de comunicación serial desarrollado por Motorola, requiere de cuatro cables MOSI (múltiples salidas con una entrada), MISO (múltiples entradas con una salida), SS (espectro encadenado) y SCLK (señal tipo reloj), se utiliza para comunicar el maestro con los dispositivos esclavos, primero el maestro configura el reloj usando la frecuencia asignada, luego selecciona el dispositivo esclavo particular para la comunicación operando el botón de selección de chip, el maestro selecciona solamente un esclavo a la vez (ver Figura 2-4) y entra en modo de comunicación full duplex. [18]

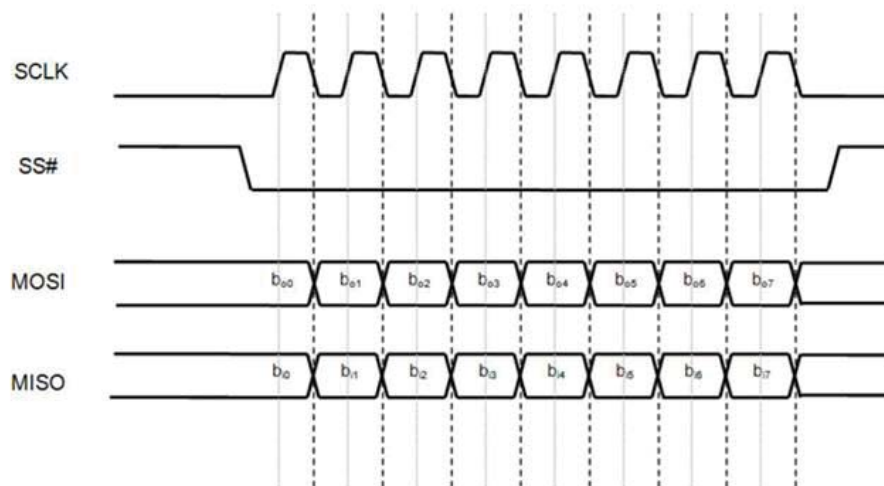


Figura 2-4 Protocolo SPI. [18]

## 2.2 Estudio y análisis de Raspberry Pi

Raspberry Pi es un ordenador de placa reducida y bajo costo, desarrollado en el Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas. Este dispositivo soporta varios componentes y puede realizar diversas funciones similares a un computador personal normal, como procesamiento de textos y datos, reproducción de videos en alta definición y conexión a Internet vía Ethernet o Wi-Fi.

## Consideraciones técnicas para mejorar el dispositivo

Siendo Raspberry Pi un ordenador de placa reducida, su software de código abierto y el sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, aunque permite otros sistemas operativos, incluido una versión de Windows 10. El diseño incluye un System-on-a-chip Broadcom BCM283x, que contiene un procesador central (CPU) ARM Cortex en versiones de 32 y 64 bits, un procesador gráfico (GPU) VideoCore IV, y 1GB de memoria RAM. El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente.

Es importante profundizar en la información que entrega el hoja de datos, para conocer los protocolos de comunicación con los que cuenta el microcontrolador de Raspberry Pi, además de la distribución que poseen los pines, identificando cuáles son los de alimentación y cuáles son los GPIO que se pueden utilizar para la transferencia de información. En la Figura 2-5 se presenta el esquemático de los pines y en la Figura 2-6 se puede visualizar las funcionalidades de cada pin.

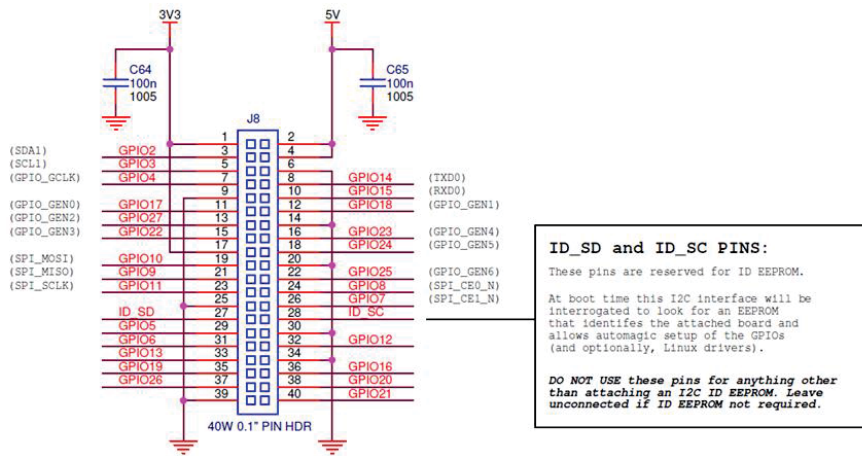


Figura 2-5 Numeración puertos GPIO Raspberry Pi. [19]

Peripherals	GPIO	Particle	Pin #	Pin #	Particle	GPIO	Peripherals			
			3.3V	1	X	X	2	5V		
I2C	GPIO2	SDA	3	X	X	4	5V			
	GPIO3	SCL	5	X	X	6	GND			
Digital I/O	GPIO4	D0	7	X	X	8	TX	GPIO14	UART	
			GND	9	X	X	10	RX	GPIO15	Serial 1
Digital I/O	GPIO17	D1	11	X	X	12	D9/A0	GPIO18	PWM 1	
Digital I/O	GPIO27	D2	13	X	X	14	GND			
Digital I/O	GPIO22	D3	15	X	X	16	D10/A1	GPIO23	Digital I/O	
			3.3V	17	X	X	18	D11/A2	GPIO24	Digital I/O
SPI	GPIO10	MOSI	19	X	X	20	GND			
	GPIO9	MISO	21	X	X	22	D12/A3	GPIO25	Digital I/O	
	GPIO11	SCK	23	X	X	24	CE0	GPIO8	SPI	
			GND	25	X	X	26	CE1	GPIO7	(chip enable)
DO NOT USE	ID_SD	DO NOT USE	27	X	X	28	DO NOT USE	ID_SC	DO NOT USE	
Digital I/O	GPIO5	D4	29	X	X	30	GND			
Digital I/O	GPIO6	D5	31	X	X	32	D13/A4	GPIO12	Digital I/O	
PWM 2	GPIO13	D6	33	X	X	34	GND			
PWM 2	GPIO19	D7	35	X	X	36	D14/A5	GPIO16	PWM 1	
Digital I/O	GPIO26	D8	37	X	X	38	D15/A6	GPIO20	Digital I/O	
			GND	39	X	X	40	D16/A7	GPIO21	Digital I/O

Figura 2-6 Especificación de funciones puertos GPIO. [20]



## Consideraciones técnicas para mejorar el dispositivo

Como se puede apreciar en la Figura 2-6, se tiene dos fuentes de alimentación de 3.3 [V] y dos de 5 [V], además de 8 GND, esto deja disponible 26 pines para puertos GPIO, de los cuales hay 12 digitales y 7 análogos. Los restantes, están dedicados a los tres protocolos de comunicación que se puede acceder por medio de los pines que son SPI, UART y I2C.

Como se mencionaba, una de las mejoras en base a la plataforma era cambiar la Raspberry Pi 2 B+ por la última versión que se encuentra en el mercado que es la Raspberry Pi 3 B+, a continuación en la Tabla 2-1 se presenta una comparación de estos dos modelos:

Tabla 2-1 Comparativa entre Raspberry Pi 2 y 3.

Placa:	Raspberry Pi 2 Model B	Raspberry Pi 3 Model B
Procesador	Broadcom BCM2836	<b>Broadcom BCM2837</b>
Núcleo CPU	Quadcore ARM Cortex-A7, 32Bit	<b>Quadcore ARM Cortex-A53, 64Bit</b>
Velocidad de reloj	900 MHz	<b>1.2 GHz</b>
RAM	1 GB	1 GB
GPU	250 MHz VideoCore IV	<b>400 MHz VideoCore IV</b>
Conexión a la red	1 x 10 / 100 Ethernet (RJ45 Port)	1 x 10 / 100 Ethernet (RJ45 Port)
Conexión inalámbrica	No tiene	<b>802.11n Wireless LAN (Wi-Fi) and Bluetooth 4.1</b>
USB Ports	4 x USB 2.0	4 x USB 2.0
GPIOs	2 x 20 Pin Header	2 x 20 Pin Header
Interfaz de cámara	15-pin MIPI	15-pin MIPI
Interfaz de monitor	DSI 15 Pin / HDMI Out / Composite RCA	DSI 15 Pin / HDMI Out / Composite RCA
Capacidad de corriente	1.8 A	<b>2.5 A</b>

Se puede apreciar que la Pi 3 presenta mejoras en el procesador, ya que pasa de 32 a 64 bits, además la velocidad de procesamiento asciende de 900 MHz a 1,2 GHz, la GPU cambio de 250 MHz a 400 MHz, también tiene incorporada una tarjeta de Wi-Fi y Bluetooth, lo que hace que sea un dispositivo mucho más poderoso pero que a la vez requiera un mayor consumo de corriente.

Si bien es cierto siempre se busca ir acorde con las nuevas incorporaciones tecnológicas, el cambio de plataforma en este caso no es necesario, ya que la plataforma requiere de conexión 3G por lo que los módulos de conexión inalámbrica no se utilizarán. Por otro lado, la plataforma E-Health no exige un procesamiento tan alto de los datos, entonces los programas correrán a la misma forma, se debe considerar también que como se cuenta con un procesador distinto que posee 64 y no 32 bits, pueden ocurrir problemas con el procesamiento de las librerías y la lectura de códigos, además el incremento de 700 [mA] en el consumo para su funcionamiento conlleva un cambio importante en la adquisición de una batería acorde a las exigencias del dispositivo.

Cabe destacar que, si a futuro se desea incorporar los protocolos de comunicación inalámbrica al proyecto, ya sea por la necesidad de realizar una conexión Wi-Fi con un router o se desea implementar la conexión Bluetooth a los sensores y la plataforma, se requerirá el cambio de la Raspberry Pi 2 a la 3 y con ello la sincronización del programa y la interfaz que ya está funcionando.

### 2.3 Adquisición de pantalla

Existen muchas versiones de pantallas que se pueden adquirir para hacer la plataforma más amigable con el usuario, a continuación, se analizarán algunas versiones y se identificará la mejor según los requerimientos del sistema.

#### 2.3.1 Adafruit HDMI 5" 800x480 Display Backpack - With Touchscreen

Se trata de un mini monitor HDMI (interfaz multimedia de alta definición) montable en panel con una pantalla táctil incorporada. Es pequeña y simple, cuenta con el TFP401 para la decodificación de vídeo, e incluye la pantalla adjunta para su plug-n-play. Tiene una resolución de 800x480, lo cual es suficiente para ejecutar la mayoría de los programas, pero lo suficientemente pequeño como para que pueda ser utilizado en proyectos portátiles o incrustado. Esta versión cuenta con capacidad de pantalla táctil, el cual permite mostrar un mouse en pantalla sin necesidad de controladores especiales, para ello se utiliza el protocolo de comunicación USB, ya que éste permite comunicar la lectura dactilar de la pantalla con la Raspberry Pi. En la Figura 2-7 se muestra la pantalla y las conexiones que requiere.



Figura 2-7 Pantalla Adafruit 5". [21]

### 2.3.2 Pi Foundation Display 7" Touchscreen

La pantalla de resolución 800x480 se conecta a través de una placa adaptadora que maneja la conversión de potencia y señal, sólo se requieren dos conexiones con la Raspberry una a partir del puerto GPIO y una conexión al puerto DSI. Los controladores dactilares de pantalla permiten el contacto con 10 puntos, además de un teclado en pantalla que se integra en el último sistema operativo Raspbian, esto permitirá obtener una funcionalidad completa sin necesidad de recurrir a un mouse ni teclado físico. La Figura 2-8 muestra la pantalla y las conexiones que requiere.



Figura 2-8 Pantalla Pi Foundation 7" Touchscreen. [22]

Existen muchas pantallas más que están adaptadas para la Raspberry Pi pero éstas son las que presentan una mayor estabilidad y son las más recomendadas por los usuarios, además cuentan con la gran ventaja de que no bloquean los pines de la Raspberry, algo que es muy importante para este proyecto ya que la Raspberry es quien controla la plataforma e-Health por lo tanto es necesario disponer de todos ellos para que el funcionamiento sea el más óptimo posible.

Dentro de estas dos versiones, existen diferencias además del tamaño, la primera es desarrollada por Adafruit y su conexión se hace por HDMI y USB siendo este último quien habilita el touchscreen, en el segundo caso la pantalla es desarrollada por la fundación Raspberry Pi, esta utiliza el DSI para comunicarse y además incluye una placa de circuito impreso a parte que hace todo el procesamiento tanto de la señal visual como del touchscreen. Dentro de estas dos opciones, destaca más la desarrollada por Raspberry ya que presenta mayor uso y recomendación de parte de los usuarios, además de que se conecta por el puerto DSI de la Raspberry permitiendo una mayor fidelidad y rapidez a la hora de operar el táctil de la pantalla.

## 2.4 Adquisición de modulo 3G

Para que el sistema se ajuste mejor a las regiones más aisladas territorialmente, es necesario implementar un tipo de comunicación que permita la transferencia de información de forma rápida y segura. Es por ello que es de suma importancia adquirir un módulo 3G, ya que es la conexión a internet más masificada en Chile. Prácticamente, en todas las zonas rurales del país existen antenas de celulares que permiten la transferencia de datos móviles. A continuación, se verán las posibilidades que existen en el mercado.

### 2.4.1 Módulo 3G / GPRS, SIM5215, adaptado por Cooking-Hacks

Este módulo 3G (Figura 2-9) funciona a una velocidad de 384Kbps de descarga y subida de datos y se conecta a través de un terminal mini-USB, específico para su ordenador portátil (Linux, Windows, MacOS). Esta especialmente orientado para trabajar con servidores de Internet e implementar internamente varios protocolos de capa de aplicación que facilitan el envío de información a la nube. Se puede usar HTTP y HTTPS (protocolo de transferencia de hipertexto) para la navegación, incluyendo la carga y descarga de contenido a un servidor web. De la misma manera trabaja con FTP y FTPS (protocolo de transferencia de archivos) protocolos que también están disponibles, éstos son muy útiles cuando la aplicación requiere el manejo de archivos. Incluso puede enviar y recibir correos electrónicos directamente desde Arduino utilizando el SMTP (protocolo para transferencia simple de correo) y POP3 clientes implementado internamente.

Su valor ronda los € 180, algo así como los \$135.000.

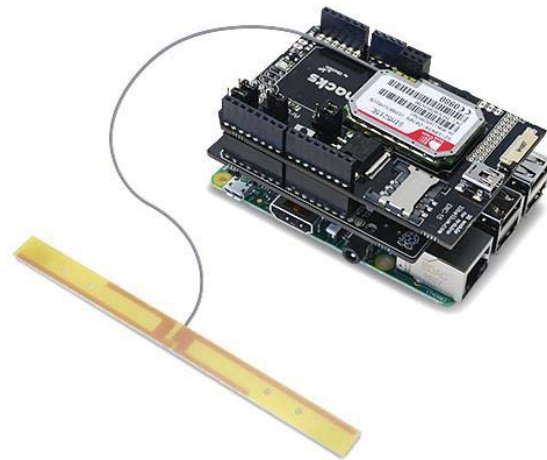


Figura 2-9 Modulo 3G Cooking Hacks. [23]

### 2.4.2 Dongle Huawei E303

El Huawei E303 (Figura 2-10) fue la primera tarjeta de datos del mundo lista para conectar y navegar, ya que no necesita ningún driver o software, la instalación, configuración y utilización son completamente gratuitas, ofrece un 75% más de velocidad que con las tarjetas de datos convencionales. Permite una conexión automática como un cable LAN, es compatible con Windows, Linux o Mac y se puede utilizar cualquier tarjeta SIM 2G/3G con 7.2 Mb/s de velocidad, HSDPA y Turbo 3G, dispone de una ranura MicroSD con una capacidad de hasta 32GB de memoria.

Su valor es de USD \$59, lo que equivale a \$40.000, incluso a veces viene de regalo con los planes de banda ancha que ofrecen las compañías telefónicas.



Figura 2-10 Dongle 3G Huawei E303. [14]

Dentro de las dos posibilidades, debemos tener en consideración la comodidad del usuario además del costo y la conexión que lleva este dispositivo. El primero se conecta vía shield que va adherido directamente a la Raspberry, ésto trae consigo un problema, ya que bloquea los pines interfiriendo en la conexión de la plataforma E-Health. Además, si se necesitara un cambio de SIM por la ausencia de cobertura, significaría una reconfiguración completa del sistema. El segundo dispositivo se conecta por USB y no necesita ningún tipo de driver ni software, no bloquea los pines de la Raspberry, también es el módulo que ya se utilizó en la iteración anterior y no habría que hacer la compra de este dispositivo. Por lo tanto, constituye la mejor opción para realizar una conexión a internet vía 3G, por lo tanto se continuará usando este dispositivo.

## 3 Creación y armado del hardware para el sistema

En esta sección se presenta el diseño y confección de la maleta que sostendrá la plataforma médica, especificando y detallando cada una de las partes diseñadas e impresas para la construcción física del dispositivo portátil.

### 3.1 Propuesta de diseño

Se propuso un sistema tipo maleta (ver Figura 3-1), que contendrá todo lo necesario para poder llegar y transportar el dispositivo. En esta, se incluirá la pantalla, Raspberry y módulo 3G en la parte superior, en la parte inferior se tendrá la plataforma E-Health, con puertos de conexión cómodos y rotulados, para evitar que el usuario conecte mal un sensor, y se produzca una lectura errónea del examen.

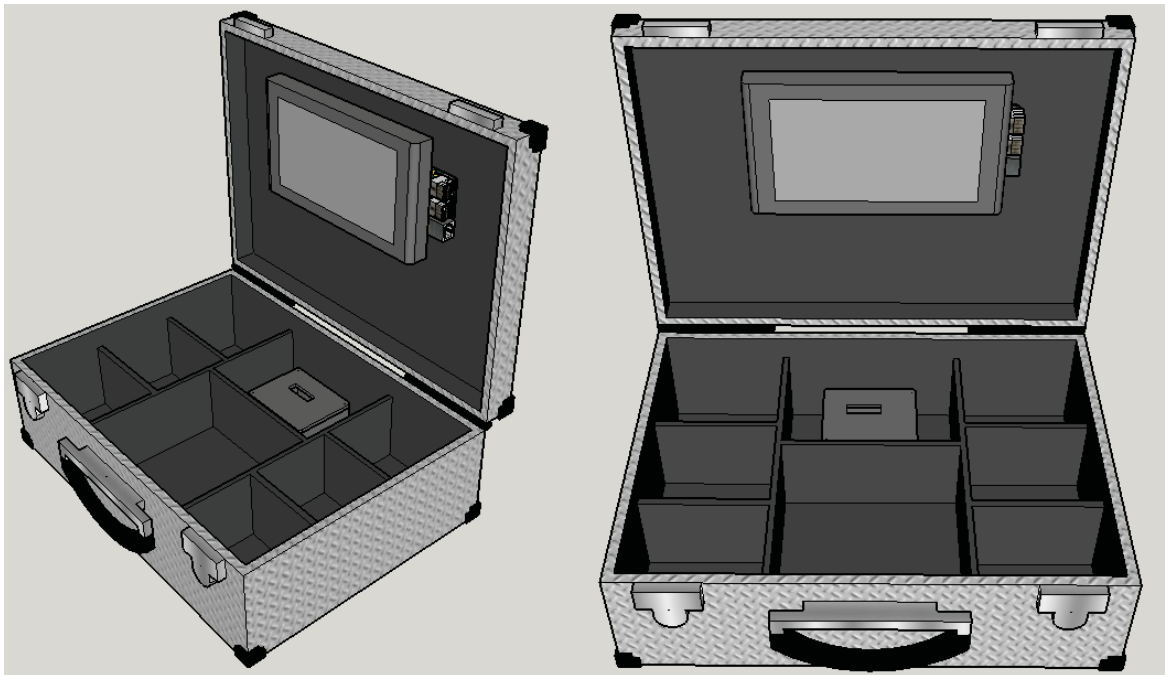


Figura 3-1 Sistema diseñado.

## 3.2 Diseños e impresiones de piezas para la maleta

La unión de los componentes y funcionamiento del dispositivo final requiere de comodidad y ergonomía a la hora de utilizarlo, por eso se diseñaron 3 piezas en 3D para facilitar la disposición y conexión de los sensores. Se diseñó una plataforma para fijar la pantalla táctil en la parte superior de la maleta, un banco de baterías para poder alimentar el dispositivo en caso de no disponer de energía eléctrica y lo más importante una caja que sostenga el shield con la plataforma E-Health para conectar los sensores.

### 3.2.1 Soporte pantalla táctil

El sistema cuenta con una pantalla táctil de 7", por lo que no se requiere del uso de un mouse para manipular la interfaz gráfica. Sin embargo, es un componente muy frágil y que funciona bajo la presión dactilar del usuario, por ende necesita un soporte firme que no se mueva al momento de ser ocupado. Además, resulta más cómodo dejar la pantalla en un ángulo de 45° que verticalmente (Figura 3-1) para manipular la interfaz a gusto.

Teniendo en consideración los aspectos mencionados anteriormente se llevó a cabo el diseño de la Figura 3-2, el cual permite elevar unos 45° la pantalla respecto al eje vertical que se crea al abrir la maleta. Cuenta con un sistema de anclaje dual para sostenerla en su posición de reposo y en la posición de uso. Cuando se ocupe de debe tomar de sus extremos y mover lateralmente a la izquierda, luego se eleva y cuando llegue a tope se regresa a su posición lateral original. De esta forma se asegura que al momento de presionar la pantalla no se suelte y baje a la posición inicial. En la Figura 3-3 se muestran las dos posiciones de la pantalla.

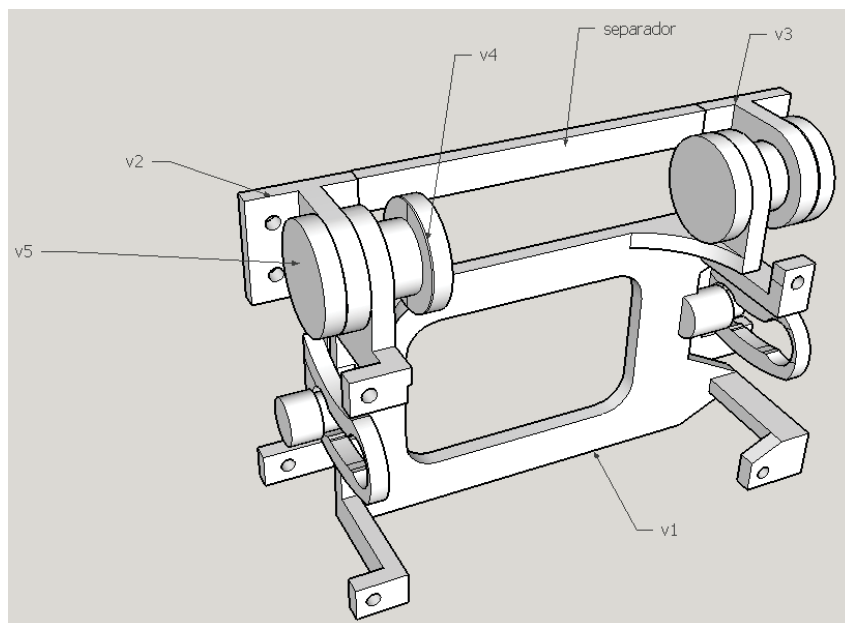


Figura 3-2 Soporte completo.



Figura 3-3 Posiciones soporte pantalla.

### 3.2.2 Banco de baterías

Para asegurar el funcionamiento de la plataforma en todo momento independiente de la disposición de energía eléctrica se creó un banco de baterías que proporciona la corriente y voltaje necesarios para el sistema, entregando una autonomía de 3 horas aproximadamente. Éste cuenta con dos pilas de níquel que están conectadas a la placa de carga la cual tiene una entrada de 5 [V] y 1[A] y una salida de 5[V] y 2 [A], además de un interruptor que permite encender y apagar el sistema. Cuando las baterías necesitan carga se enciende una luz celeste intermitente la que queda fija una vez que estén cargadas.

Todo este sistema se introdujo dentro de una caja diseñada en 3D (ver Figura 3-4) que dispone de la abertura para la conexión del cargador a la placa y el interruptor. En la Figura 3-5 se puede apreciar la disposición del banco de baterías en la maleta.

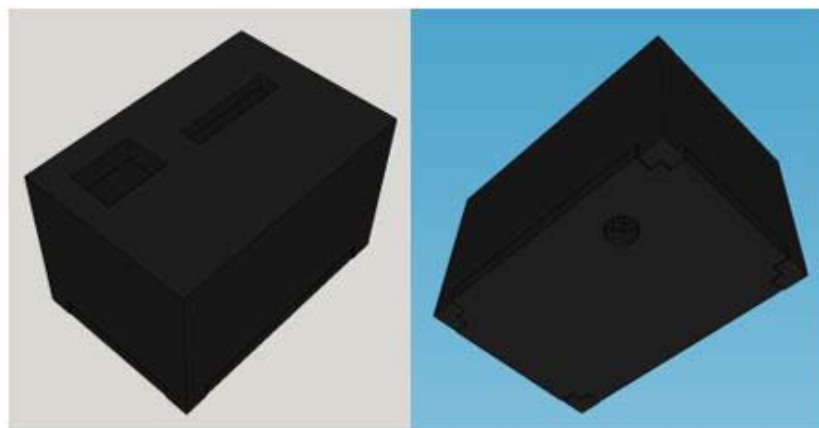


Figura 3-4 Caja banco de baterías diseñado en 3D.





Figura 3-5 Banco de baterías.

### 3.2.3 Caja para plataforma E-Health

La conexión de cada sensor debe realizarse de forma cómoda y segura. En las primeras iteraciones, ésta se realizaba a la plataforma E-Health directamente sin tener opciones intermedias de conectividad ni protecciones. En esta iteración cada sensor cuenta ya sea con una conexión Jack o USB que permite la obtención de los datos desde los sensores hacia la plataforma.

Se creó una caja portadora con la finalidad de indicar las posiciones de cada sensor y exponer lo menos posible la plataforma que es el cerebro del sistema. En la Figura 3-6 se presenta el diseño en 3D de la caja contenedora y en la Figura 3-7 se presenta implementada en la maleta. Se denota una guía de cables de colores que van desde la pantalla hacia la caja, estos conectan la Raspberry Pi con el shield de la plataforma E-Health.

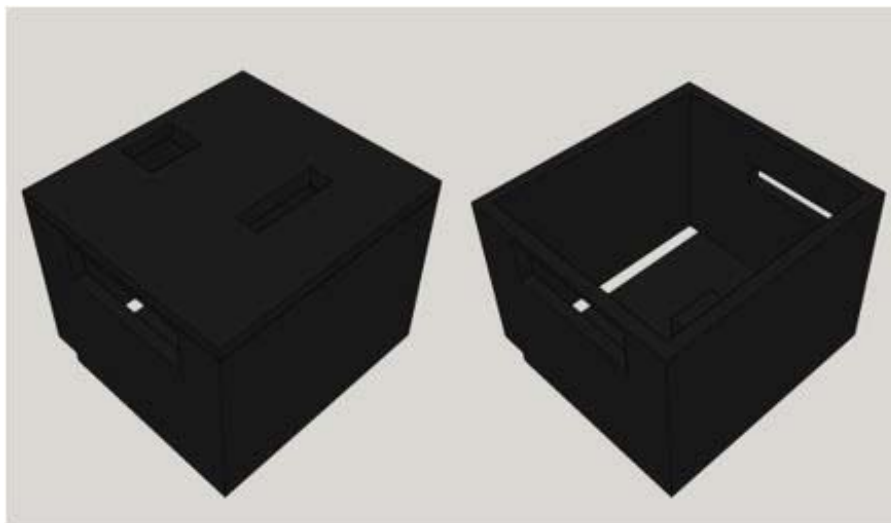


Figura 3-6 Caja para shield diseñada en 3D.

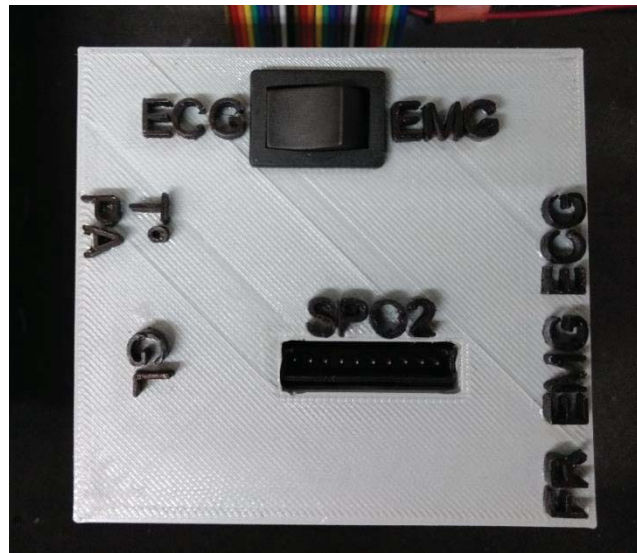


Figura 3-7 Caja portadora.

### 3.3 Resultado final

En la Figura 3-8 se puede observar el resultado final de la maleta, el cual se consiguió luego de varios ajustes a los diseños que se realizaron para el soporte de la pantalla, el banco de baterías y la caja contenedora del shield. La maleta cuenta con 8 subdivisiones las cuales permiten almacenar los sensores por separado y entregarle el espacio necesario a la caja contenedora y el banco de baterías. En la parte izquierda se encuentran los sensores de flujo respiratorio, pulso y oxígeno y glucosa, en la parte central el sensor de presión y temperatura, al lado derecho los electrodos del electrocardiograma y electromiograma que disponen de una conexión Jack al igual que el de flujo respiratorio.



Figura 3-8 Dispositivo portátil.

### 3.4 Calibración de sensores gráficos

En esta sección se muestra cual fue el proceso de calibración de los sensores gráficos de la plataforma E-Health, partiendo por una falla que existía en la toma de datos de dos exámenes (ver Figura 3-9) donde las gráficas de el electrocardiograma y electromiograma eran completamente erróneas y se medía solo ruido.

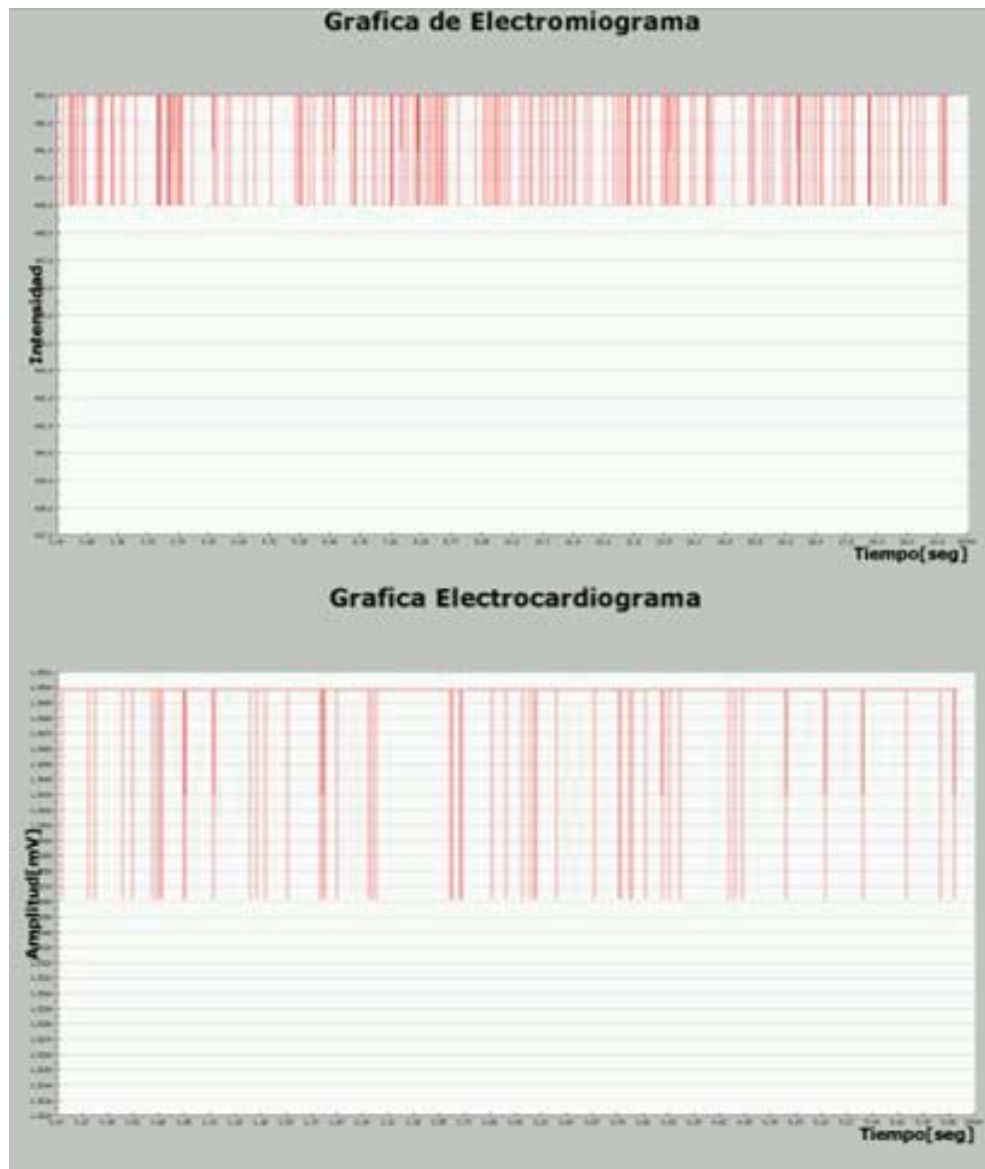


Figura 3-9 Gráficas obtenidas antes de la calibración.

Una de las modificaciones que se hizo fue en la librería de la plataforma, específicamente en el archivo ehealth.cpp, donde se modificó la línea de la función de obtención de datos del electrocardiograma. Particularmente en el retorno de la lectura análoga, se cambió la exactitud de datos que se mapeaba, ya que se cambió el multiplicador de la variable float de un 5 a un 3.5 tal como se muestra en la Figura 3-10.

```

//!*****
//!   Name:   getECG()                               *
//!   Description: Returns an analogic value to represent the ECG. *
//!   Param : void                                   *
//!   Returns: float with the ECG value in voltage *
//!   Example: float volt = eHealth.getECG();      *
//!*****

float eHealthClass::getECG(void)
{
    float analog0;
    // Read from analogic in.
    analog0=analogRead(0);
    // binary to voltage conversion
    return analog0 = ((float)analog0 * 3.3) / 1023.0;
}

//!*****
//!   Name:   getEMG()                               *
//!   Description: Returns an analogic value to represent the EMG. *
//!   Param : void                                   *
//!   Returns: float with the EMG value in voltage *
//!   Example: float volt = eHealth.getEMG();      *
//!*****

int eHealthClass::getEMG(void)
{
    int analog0;
    // Read from analogic in.
    analog0=analogRead(0);
    // binary to voltage conversion
    return analog0;
}

```

Figura 3-10 Código de la librería de la plataforma.

También es de suma importancia utilizar un switch para realizar estos dos exámenes, ya que como usan la misma entrada análoga para realizar la toma de datos, ésta debe cambiarse a los pines correspondientes donde se conectarán los electrodos. La Figura 3-11 muestra la disposición del switch en la caja contenedora del shield.

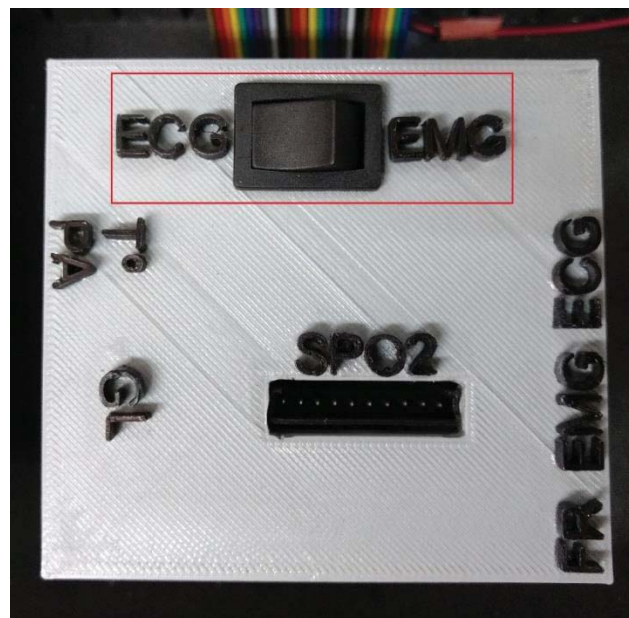


Figura 3-11 Switch que permite cambiar los pines para la obtención de datos de los exámenes.

Finalmente, los gráficos obtenidos son como se muestran en la Figura 3-12, donde se pueden apreciar perfectamente los pulsos del electrocardiograma y los esfuerzos musculares leídos por electromiograma.

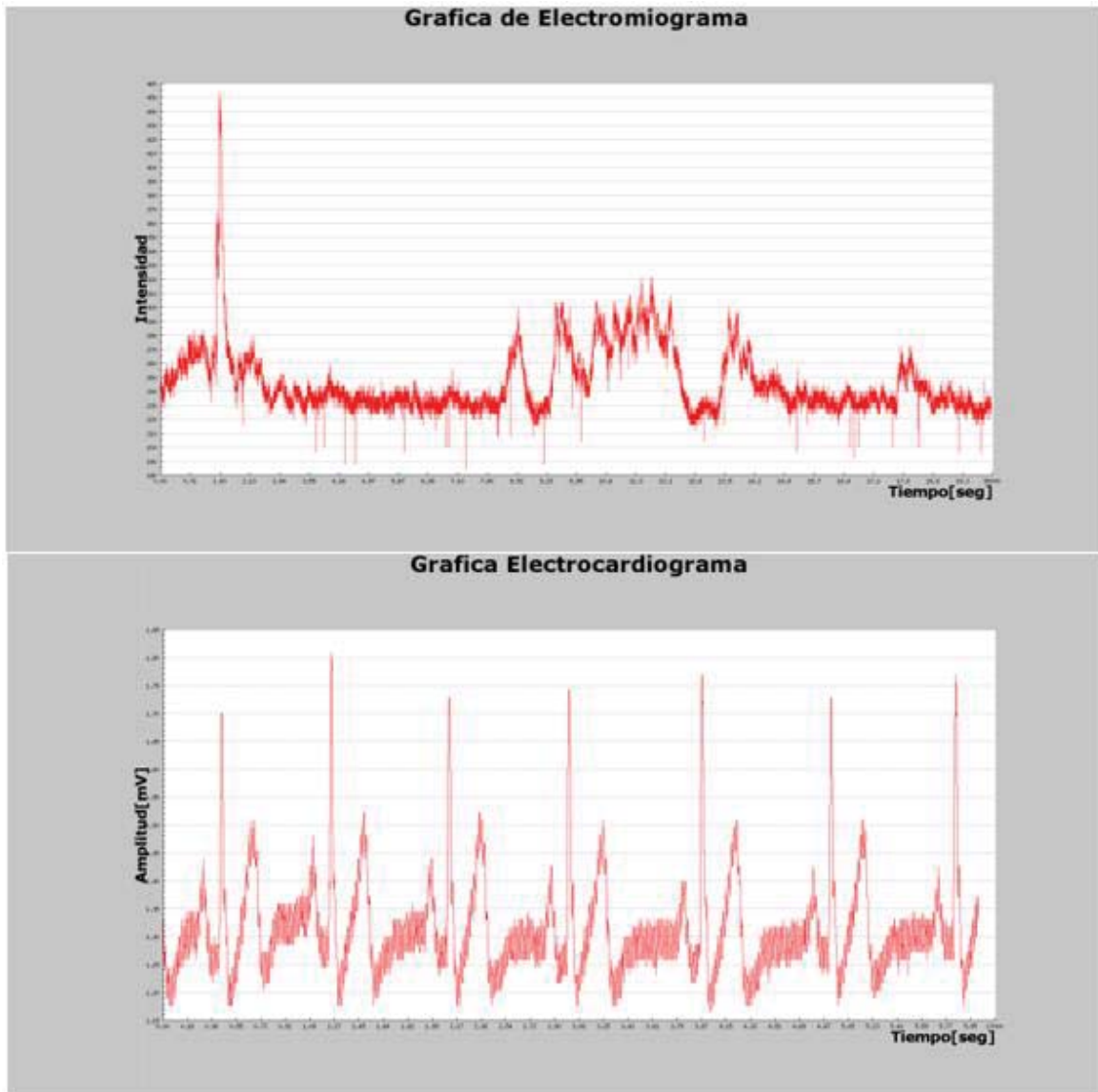


Figura 3-12 Exámenes gráficos mejorados.

## 4 Modificación y actualización de software

En esta sección se explica el proceso realizado para mejorar el funcionamiento y aspecto de la interfaz gráfica del sistema de telemedicina, ya que contaba con una presentación muy sencilla, que no presentaba las especificaciones necesarias para que cualquier persona lo pudiese utilizar.

### 4.1 Proceso de desarrollo de la interfaz gráfica

A continuación, se explica que es una interfaz gráfica y porque es tan relevante su implementación en este proyecto. También se comenta el tipo de lenguaje en el que se encuentra desarrollada, dando sus principales ventajas y desventajas respecto a otras opciones.

#### 4.1.1 ¿Qué es una interfaz gráfica?

La interfaz gráfica de usuario (Figura 4-1), es un programa que actúa como comunicador entre la máquina y el usuario, por medio de un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo, manipulable y cómodo que permita al usuario ejecutar todas las acciones disponibles por la máquina. [24] En este proyecto, es de vital relevancia contar con una comunicación directa entre el personal médico y la plataforma electrónica que obtiene los datos, ya que con ello se facilita el uso y la obtención del diagnóstico para el paciente. Lo más importante de esta aplicación se da en el caso de los exámenes que realizan gráficas, como el electrocardiograma, electromiograma y flujo respiratorio, debido a que permite visualizar de forma inmediata los datos obtenidos.



Figura 4-1 Interfaz gráfica.

### 4.1.2 Lenguaje utilizado

La creación de la interfaz gráfica para la plataforma E-Health debe estar enfocada en la facilidad para conectar el dispositivo vía web y realizar la transferencia de datos de forma directa con un servidor. Asegurando que el personal médico pueda enviar y revisar desde cualquier lugar los resultados obtenidos por el dispositivo.

Lenguajes como Python y C ++, que son ampliamente utilizados para el desarrollo de interfaces gráficas en los dispositivos no permiten la derivación de la misma interfaz gráfica vía web. Esto si se puede realizar con el conjunto de lenguajes PHP y HTML por medio de un servidor web APACHE, permitiendo cumplir la función de conectividad que requiere este proyecto.

## 4.2 Modificaciones a la interfaz gráfica

La interfaz gráfica contaba con 6 ventanas que permitían al usuario ingresar su perfil, datos del paciente, realizar las mediciones, anotar observaciones y crear el reporte médico para así enviarlo vía correo electrónico. Ahora se realizó una actualización con 7 ventanas, que contempla las mencionadas anteriormente, más una ventana de instrucciones de uso, que permite al usuario recurrir a la información en cualquier momento.

### 4.2.1 Implementaciones nuevas

Cabe destacar que el sistema anterior contaba con alrededor de 60 archivos que permitían crear la interfaz gráfica, lo cual hacía que el sistema fuera muy lento y tardara tanto en el cambio de ventana como en la obtención de los datos. Actualmente eso se redujo a 40 archivos que provocaron un mejoramiento en el comportamiento visual y funcional de todo el sistema.

Se creó un estilo genérico mediante un archivo CSS (hojas de estilo en cascada). Este tipo de archivos representa hojas de estilo en cascada (Cascading Style Sheets) y es un lenguaje utilizado para especificar la presentación de documentos en una página web. Los elementos dentro del documento CSS afectan a la apariencia, color, fuente, imágenes y otros detalles. Además, permite la gestión mejorada del contenido, flexibilidad y accesibilidad a la información de la página web. El uso de CSS posibilita que varias páginas cuenten con el mismo formato y la misma organización fácilmente, así como utilizar métodos de renderización. [25]

El uso de esta herramienta permitió reducir la cantidad de archivos que conforman la interfaz y simplificar el código de cada uno de ellos. A continuación, en la Figura 4-2 y la Figura 4-3 se presenta un ejemplo y se hace una comparación del archivo principal.php que contiene el código de la primera ventana que se despliega, en donde se escribe el usuario y contraseña. En la Figura 4-2 se visualiza el código antiguo y específicamente desde la línea 33 hasta la 71 está la configuración de los botones y el estilo de letra. A diferencia de la Figura 4-3 que ya en la línea 30 cita el archivo estilo.css que contiene toda la configuración predeterminada.

```

21 <?php
22 $name=$_POST["name"];
23 $pass=$_POST["pass"];
24 if(!isset($name) && !isset($pass))
25 {
26     ?>
27     <html>
28     <body background="fondo.jpg">
29     <meta charset="utf-8">
30     <font color="black" size=4 face="arialblack">
31     <h1 align="center">Tecllea un Usuario y contraseña</h1>
32     <style type="text/css">
33     .boton{
34         font-size:17px;
35         font-family:Verdana,Helvetica;
36         font-weight:bold;
37         color:white;
38         background:#638cb5;
39         border:10px;
40         width:200px;
41         height:30px;
42     }
43     .boton1{
44         font-size:17px;
45         font-family:Verdana,Helvetica;
46         font-weight:bold;
47         color:white;
48         background:#638cb5;
49         border:0px;
50         width:25px;
51         height:25px;
52     }
53

```

Figura 4-2 Antiguo pricipal.php

```

18 <?php
19 $name = $_POST["name"];
20 $pass = $_POST["pass"];
21 if (!isset($name) && !isset($pass)) {
22     ?>
23     <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
24     <html lang="es">
25     <head>
26         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
27         <meta http-equiv="X-UA-Compatible" content="IE=edge">
28         <meta name="viewport" content="width=device-width, initial-scale=1">
29         <title>Plataforma E-Health</title>
30         <link href="css/estilo.css" rel="stylesheet">
31         <!--<link href="css/style.css" rel="stylesheet">-->
32         <link href="https://fonts.googleapis.com/css?family=Ubuntu" rel="stylesheet">
33         <link rel="icon" href="src/favicon.png">
34         <meta name="viewport" content="width=device-width, initial-scale=1.0">
35         <script type="text/javascript" src="js/jquery-1.7.1.min.js"></script>
36         <link href="js/jquery-ui.css" rel="stylesheet" />
37         <script type="text/javascript" src="js/jquery.min.js"></script>
38         <script type="text/javascript" src="js/jquery-ui.min.js"></script>
39         <link href="css/keyboard1.css" rel="stylesheet" />
40         <link href="css/estilos.css" rel="stylesheet" />
41         <script type="text/javascript" src="js/jquery.keyboard.js"></script>
42         <script src="js/jquery.keyboard.extension-typing.js" type="text/javascript"></script>
43         <script type="text/javascript">
44             $(document).ready(function () {
45                 $('#txtkeyboard1').keyboard({
46                     autoAccept: true
47                 })
48             })
49             $('#txtkeyboard').keyboard({
50                 autoAccept: true
51             })
52         })
53         $('#txtkeyboard2').keyboard({

```

Figura 4-3 Nuevo principal.php.



## 4.2.2 Comparación de las modificaciones realizadas

A continuación, se presentan las modificaciones realizadas a cada una de las ventanas que conforman la interfaz gráfica del sistema. Al encender el dispositivo portátil se ejecuta de forma automática la interfaz gráfica a pantalla completa, quedando a disposición del usuario la plataforma de medición.

En la Figura 4-4 parte superior se muestra la interfaz de bienvenida anterior y en la parte inferior la versión nueva. En ambos casos para acceder a la siguiente ventana se debe ocupar el nombre de usuario “user1” y la contraseña “pass1”.



Figura 4-4 Comparación de interfaz inicial.

Una vez escrita la información, se despliega la ventana en donde se deben ingresar los datos del paciente y el medico a cargo especificando nombre, run edad y el lugar donde se está realizando la medición. Esto se muestra en la Figura 4-5, donde en la parte superior e inferior se muestra la ventana antigua y nueva respectivamente.



Figura 4-5 Comparación interfaz de datos del paciente.

Luego de que todos los datos estén ingresados correctamente se despliega el interfaz de Menú Principal, es acá donde se mostrarán los datos de las diferentes mediciones realizadas por los sensores. En esta interfaz se visualizarán todas las muestras tomadas por el sensor de pulso y oxígeno, temperatura, glucómetro y presión arterial. Además, se ve la sección de los exámenes que requieren una interpretación gráfica como el flujo respiratorio, electrocardiograma y electromiograma. En la parte derecha se despliegan opciones como observaciones, reporte médico, borrar mediciones y cierre de sesión que facilitan el uso de esta plataforma.

En la Figura 4-6 se realiza una comparación a las modificaciones que se realizaron al menú principal, que a pesar de que no varía en gran medida, se mantiene el mismo estilo que se está utilizando en la nueva versión. Cabe destacar que una actualización con otro formato mucho más organizado dejaba inoperativo al sistema, por lo que esta ventana es la que sufre menos cambios.



Figura 4-6 Comparación interfaz menú principal.



Figura 4-7 Comparación ventana Gráficas.

Las mediciones que se representan gráficamente son tomadas en otra ventana, la cual también se modificó y se puede comparar en la Figura 4-7. Ésta se despliega una vez que se presiona el botón Graficas en el menú principal y cuenta con 3 botones que son Respiración, Electrocardiograma y Electromiograma, los cuales derivan al gráfico de su respectivo examen.

En la parte derecha del menú principal, se tiene el botón observaciones el cual deriva a otra ventana la cual se muestra y compara en la Figura 4-8. Ésta permite anotar cualquier tipo de anomalía o característica que presenten los exámenes realizados.



Figura 4-8 Comparación ventana observaciones.

En la parte derecha, también se encuentra el botón instrucciones, el cual es nuevo ya que en la iteración no se consideró poner el detalle de cómo utilizar cada sensor, lo cual es de suma relevancia en el caso de que el usuario no cuente con ninguna capacitación previa. En la Figura 4-9 se muestra la ventana que contiene las instrucciones de como conectar y realizar todos los exámenes, cuenta con una imagen en cada caso que permite hacer mucho más didáctico el proceso y con un botón que permite regresar al menú principal. Cabe destacar que esta se puede utilizar en cualquier momento y no afecta las mediciones.



Figura 4-9 Ventana instrucciones.

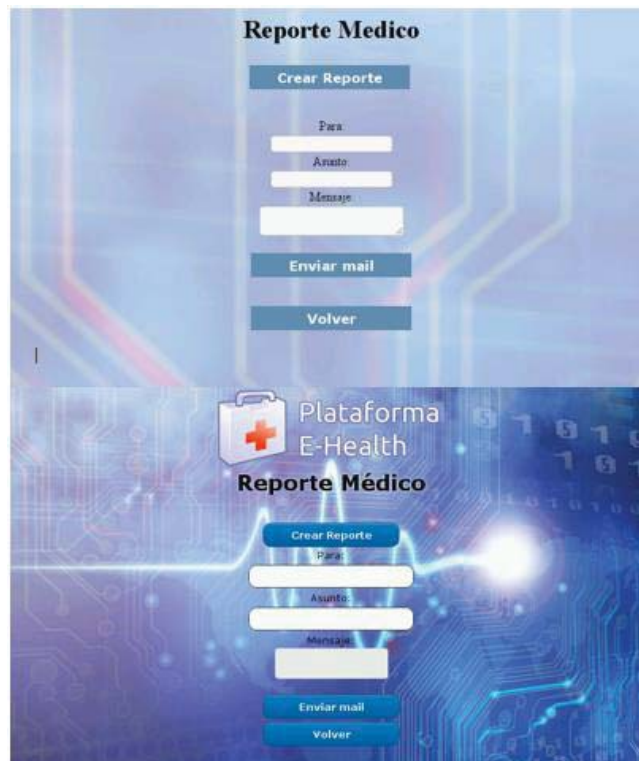


Figura 4-10 Comparación ventana reporte médico.

Bajo el botón observaciones se encuentra reporte médico, que permite acceder a la parte final del proceso, donde al elegir la opción crear reporte se realizara un documento PDF que contiene todas las mediciones y los datos del paciente. Esto se muestra y compara en la Figura 4-10 donde en la parte superior se encuentra la versión antigua y en la inferior la nueva.

Finalmente, en la Figura 4-11 se muestra la remodelación de la ventana borrar mediciones, la cual corresponde al último botón en la parte derecha del menú principal y permite borrar todos los datos tomados con anterioridad.



Figura 4-11 Comparación ventana borrar mediciones.

# 5 Creación e implementación de una base de datos

La creación de una base de datos dentro del software requiere de la instalación de varios programas que permiten crear y manejar las mediciones en el sistema, con el fin de almacenar la información de los exámenes realizados. Programas como Apache2, MySQL, phpMyAdmin, ayudan a formar una base de datos cómoda y amplia que satisfacen por completo los requerimientos de esta nueva modificación.

## 5.1.1 ¿Qué hacen los programas utilizados?

Apache corresponde a un servidor web HTTP de código abierto que permite implementar este protocolo y la noción de sitio virtual, es mantenido y desarrollado por una comunidad de usuarios bajo la supervisión de Apache Software Foundation. Es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web y en este caso se utiliza como servidor local con el fin de previsualizar y probar los códigos PHP mientras son desarrollados. [26]

MySQL es uno de los sistemas de gestión de base de datos open source más popular del mundo para desarrollo web. Comprada y últimamente desarrollada por Oracle Corporation, con un uso principal como sistema relacional de datos, que permite archivarlos en tablas por separado en vez de colocarlos en un gran archivo como lo haría Microsoft SQL Server. Esto permite aumentar la velocidad y flexibilidad a la hora de requerir la información almacenada en el servidor. [27]

PHPMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de una página web utilizando internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos. [28]

Estos tres programas serán utilizados para crear la base de datos que almacenará la información del paciente, médico y los exámenes de presión, saturación de oxígeno en la sangre, pulso, temperatura corporal, glucosa en la sangre. Con lo cual se podrá generar un registro con la fecha exacta en que el paciente se tomó las mediciones, almacenando los datos de cada sesión ejecutada.

## 5.1.2 Instalación de los programas

La instalación de los programas en la Raspberry Pi se hace mediante el uso de LXTerminal que corresponde al ejecutor de comandos del sistema Raspbian. Para ello se debe ejecutar los siguientes comandos:

```
pi@raspberrypi~$ sudo apt-get update
pi@raspberrypi~$ sudo apt-get upgrade
```

El primer comando permite actualizar la lista de paquetes disponibles y sus versiones, pero no instala o actualiza ningún paquete. Esta lista la toma de los servidores con repositorios que están definidos en la sources.list de la versión de Raspbian que se disponga. El segundo comando se ejecuta una vez que el primero descargó la lista de softwares disponibles, actualiza esos paquetes e instala las nuevas versiones respetando la configuración del software en la mayor medida.

Luego se ejecutan los comandos de instalación de MySQL y Apache:

```
pi@raspberrypi~$ sudo apt-get install mysql-server
pi@raspberrypi~$ sudo apt-get install apache2 php5 libapache2-mod-php5
```

Estos paquetes ya estaban instalados, por lo que los dos programas sólo se actualizaron y no se requirió hacer nada adicional. Luego se ejecuta el comando mostrado a continuación para instalar phpMyAdmin y se sigue el proceso completo ya que éste no se encontraba en la memoria de la Raspberry Pi .

```
pi@raspberrypi~$ sudo apt-get install phpmyadmin
```

Una vez descargados los archivos se continua con la configuración correspondiente mostrada en la Figura 5-1 .

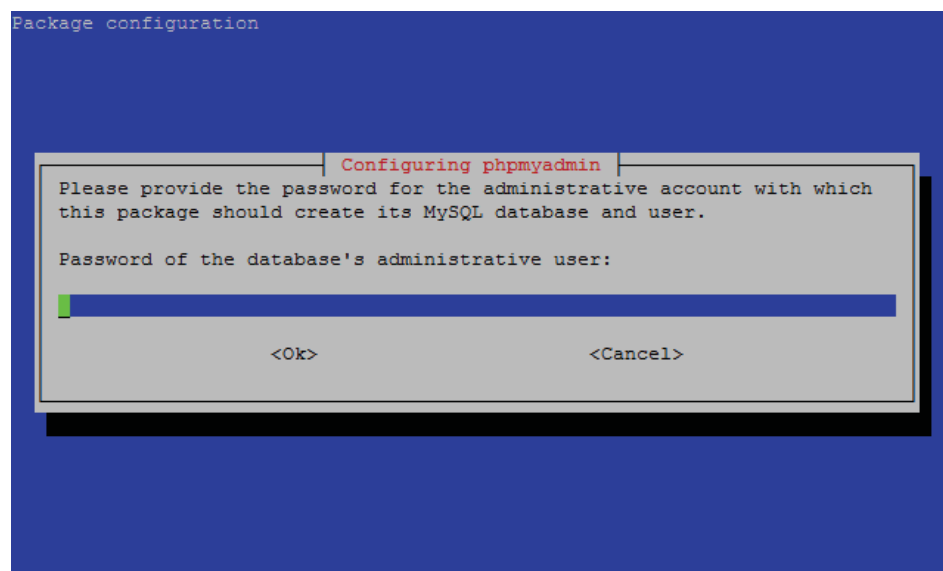


Figura 5-1 Configuración password phpMyAdmin.



Se ejecuta el comando mostrado a continuación y se modifica el archivo de la figura Figura 5-2 añadiéndole `Include /etc/phpmyadmin/apache.conf` al final del texto.

```
pi@raspberrypi~$ sudo nano /etc/apache2/apache2.conf
```

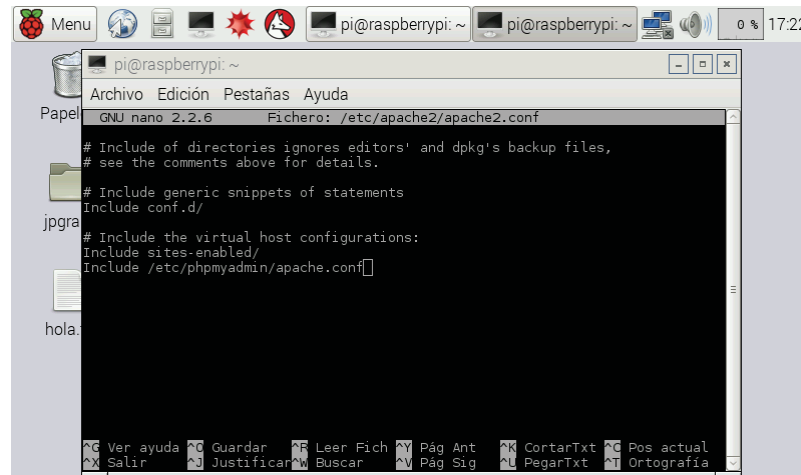


Figura 5-2 Modificación de apache2.conf.

Por último, se escribe el ultimo comando con el fin de reiniciar el servidor Apache e incorporar el phpMyAdmin en el uso de la base de datos.

```
pi@raspberrypi~$ sudo /etc/init.d/apache2 restart
```

Es así como quedan instalados todos los programas que permitirán hacer un manejo de datos de los pacientes, doctores y exámenes además contar con un registro que guarde la información de todas las muestras realizadas a lo largo del tiempo.

## 5.2 Programación y funcionamiento de la base de datos

La creación de una base de datos ayuda a almacenar en el tiempo, los datos de los pacientes para poder acceder a un historial médico de exámenes. Esto se elabora mediante la creación de una base de datos en MySQL, la que contiene 3 tablas de información: doctores, pacientes y exámenes (ver Figura 5-3). Para crear este entorno de almacenamiento se utiliza el software web phpMyAdmin, cuya distribución es gratuita y su interfaz muy fácil de manejar.



Figura 5-3 Diagrama de las tablas realizadas.

## Creación e implementación de una base de datos

Para poder visualizar las tablas en la Raspberry Pi se debe ingresar en el servidor local a phpMyAdmin, que aparecerá como en la Figura 5-4, donde se tiene que poner como contraseña “pass”. Cuando ya se ingresa a la aplicación, aparece ehealth\_bd que corresponde al conjunto de tablas (base de datos creada), ahí se encontraran las tres subdivisiones mencionadas anteriormente tal como aparece en la Figura 5-4.

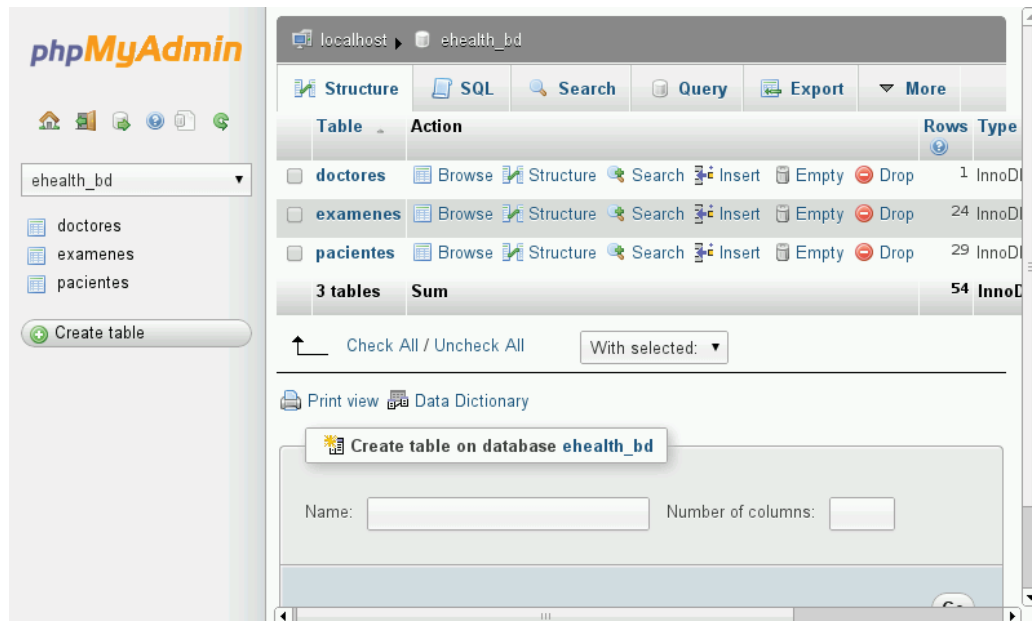


Figura 5-4 Estructura de ehealth\_bd.

La tabla de doctores (ver Figura 5-5), tiene información de las cuentas permitidas en el sistema y sus usuarios solo podrán ser editados por el administrador del sistema.

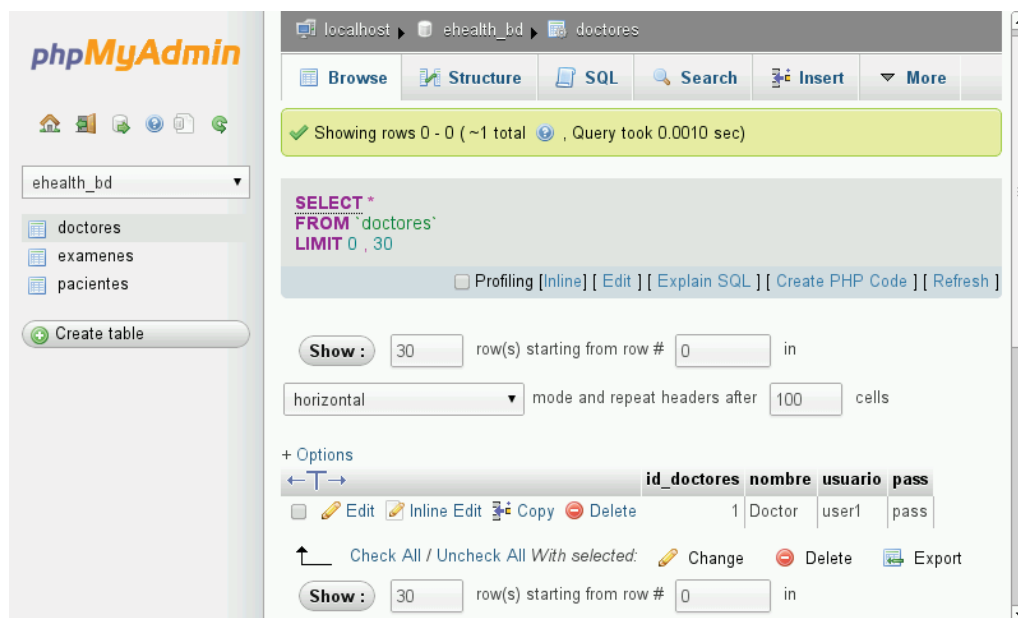


Figura 5-5 Tabla doctores.

Cada entrada de la tabla de pacientes (ver Figura 5-6), es una ficha de un paciente, la cual contiene datos muy importantes como por ejemplo el nombre completo, RUT, edad, procedencia y además vincula cada paciente con la ID del doctor que creó la ficha y realizó los exámenes. El límite de esta base de datos es de 30 pacientes, luego de superado ese valor se van limpiando los registros más antiguos, con la finalidad de no colapsar el servidor local. En caso de disponer de un servidor web para subir los archivos, se puede eliminar la restricción y subir datos de pacientes ilimitadamente.

The screenshot displays the phpMyAdmin interface for the 'ehealth\_bd' database, specifically the 'pacientes' table. The table structure and data are as follows:

id_pacientes	nombre	rut	edad	servicio	procedencia	medico
1	Esteban Perez	18561579-0	24	CONSULTORIO	VALPARAISO	Doctor
2	Esteban Perez	18561579-0	24	CONSULTORIO	SALAMANCA	Doctor
3	Esteban Perez	18561579-0	24	CONSULTORIO	SALAMANCA	Gloria Campos
4	JULIETA PALACIOS	10501160-1	49	CONSULTORIO	SALAMANCA	Gloria Campos
5	GRACE ROJAS	15047780-8	33	CONSULTORIO	LLIMPO	Gloria Campos
6	MABEL PALACIOS	10501168-7	51	CONSULTORIO	SALAMANCA	Doctor

Figura 5-6 Tabla de pacientes ingresados al sistema.

Cada entrada de la tabla exámenes (ver Figura 5-7) está asociada a una ficha de paciente mediante su ID, también con la ID del médico que hizo el examen, además se genera una propia ID que permite separar los exámenes realizados y ordenarlos. Los datos mostrados en la tabla corresponden a los sensores de presión arterial (presión sistólica y diastólica), pulso y oxígeno en la sangre, temperatura corporal y glucosa en la sangre. Todos estos datos aparecerán una vez que se busque el paciente por su RUT en la interfaz gráfica del sistema.

The screenshot shows the phpMyAdmin interface for a database named 'ehealth\_bd'. The 'examenes' table is selected, and a SQL query is executed: `SELECT * FROM examenes LIMIT 0, 30`. The table displays 6 rows of data. Below is a representation of the data shown in the table:

	id_examenes	id_pacientes	id_doctores	diastolica	sistolica	temperatura	glucosa	saturacion	pulso	observaciones	fecha
<input type="checkbox"/>	1	1	1	111	74	35.81	79	98	69	Normal	2017-11-30
<input type="checkbox"/>	2	2	1	115	71	36.99	80	97	90	Normal	2017-12-06
<input type="checkbox"/>	3	3	1	128	81	35.79	90	97	76	Normal	2017-11-30
<input type="checkbox"/>	4	4	1	135	80	36.37	104	96	72	Estado normal, a considerar presion un poco alta.	2017-12-06
<input type="checkbox"/>	5	5	1	107	74	36.83	103	98	70	Estado normal, todos los signos vitales en orden.	2017-12-06
<input type="checkbox"/>	6	6	1				94	86			2017-12-06

Figura 5-7 Tabla con almacenamiento de datos de los exámenes tomados.

Finalmente, por medio de estas tablas creadas se puede exportar todo lo realizado en phpMyAdmin a código SQL, con la finalidad de poner trasladar todo el trabajo realizado a otra base de datos o a otro servidor, en los que se quiera replicar el trabajo.

### 5.3 Resultado y visualización de la base de datos en la interfaz gráfica

En esta sección, se explican los cambios de la interfaz gráfica que introduce la presencia de una base de datos, la que crea un registro de todos los exámenes médicos tomados. La única ventana modificada es aquella donde se introducen los datos del paciente, que se despliega una vez que se accede a la cuenta. En este caso, el médico tiene la opción de realizar un nuevo examen o revisar el resultado de los exámenes que se realizaron con anterioridad.

#### 5.3.1 Visualización gráfica de la nueva incorporación

En la Figura 5-8 se observa el cambio de la ventana de ingreso de datos que ahora cuenta en la parte inferior con un buscador de exámenes que al ingresar el Rut de un paciente muestra los resultados obtenidos de los exámenes de pulso, saturación de oxígeno, presión, glucosa en la sangre y temperatura corporal.

The screenshot shows a web interface for 'Plataforma E-Health' with a medical bag icon. The main heading is 'Datos del Paciente'. The form contains several input fields with pre-filled text: 'Nombre: Esteban Perez', 'RUT: 18561579-0', 'Edad: 24', 'Servicio: CONSULTORIO', 'Procedencia: LLIMPO', and 'Médico: Doctor'. There is a blue 'Ingresar Datos' button. At the bottom, there is a search section with 'Búsqueda de Paciente por RUT: 18561579-0' and a blue 'Búsqueda' button.

Figura 5-8 Ventana modificada para el ingreso a la base de datos.

Una vez ingresado el RUT, se despliega otra ventana que mostrará el registro asociado a éste (Figura 5-9). En éste se presenta una lista de máximo 10 exámenes por página que a medida que se van acumulando se crean más pestañas para mantener en orden los datos.

The screenshot shows the 'Resultados de Búsqueda' page. It features a search bar at the top with the RUT '18561579-0' and a 'Búsqueda' button. Below the search bar, there is a list of search results. The first result is '30/11/2017 Doctor Ver Examen'. The second result is '06/12/2017 Doctor Ver Examen'. The third result is '30/11/2017 Doctor Ver Examen'. There are navigation arrows '< 1 >' and a 'Volver' button at the bottom.

Figura 5-9 Ventana con la base de datos.

Al presionar, "Ver Examen" se encuentra toda la información que se guardó (Figura 5-10), como el nombre del paciente y del doctor, además de los resultados del examen de glucosa en la sangre pulso y saturación de oxígeno, presión arterial y temperatura corporal.

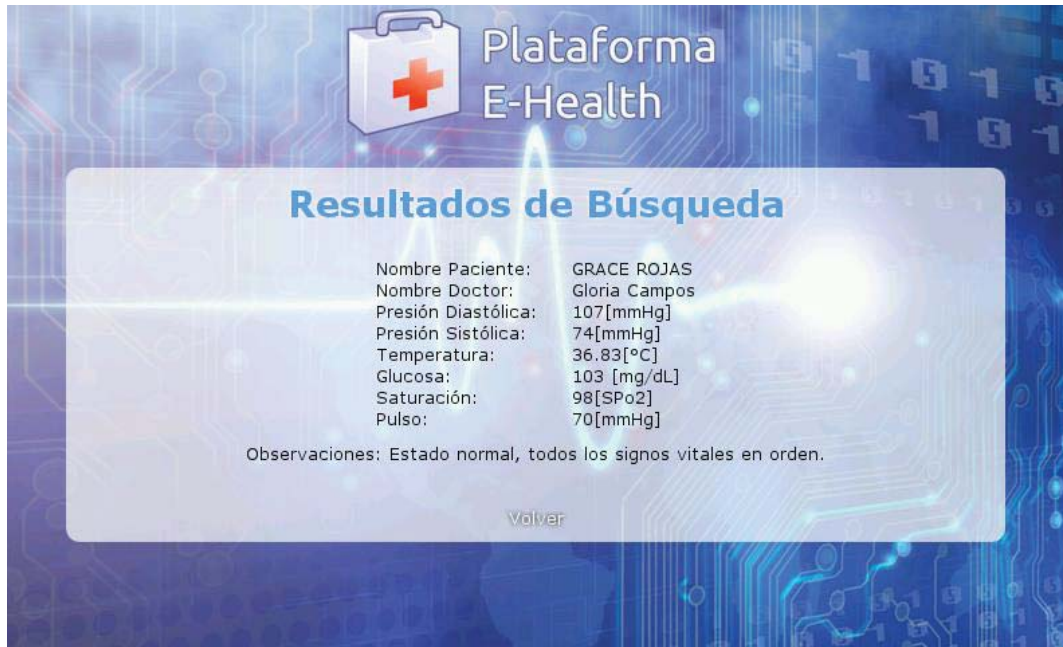


Figura 5-10 Datos almacenados en el registro de pacientes.

Esto nos permite por tanto almacenar la información de cada vez que se ingrese un paciente al sistema y sea examinado por el medico a cargo. También es una herramienta fundamental para hacer un seguimiento de las condiciones y progreso que posee la persona a lo largo de todo el tiempo que ha sido atendida.

## 6 Resultados finales y comparación de mediciones

En esta sección se presentan los resultados obtenidos por el dispositivo portátil en un chequeo completo a la paciente Julieta Palacios en el Consultorio del poblado de Llimpo en la comuna de Salamanca. Se contrastan las mediciones realizadas por la plataforma E-Health versus las tomadas por los diferentes sensores con los que disponía el consultorio.

### 6.1 Comparación de exámenes realizados

A continuación, se compararán una por una las mediciones de cada sensor, con el fin de verificar la efectividad de la plataforma E-Health y observar algún tipo de inconveniente para así mejorarlo en caso de que sea necesario.

#### 6.1.1 Temperatura corporal

En el lado izquierdo de la Figura 6-1 se presenta el valor de 36.8 °C arrojado por el termómetro de mercurio que se utiliza en el consultorio y en el lado derecho se muestra el valor tomado por un sensor de temperatura que utiliza el infrarrojo para verificar de una forma rápida si es que el paciente presenta un grado de fiebre, Éste arroja un valor de 37.1°C.



Figura 6-1 Medición termómetros propios.

El valor entregado por la plataforma E-Health fue de 36.37°C , este valor se puede visualizar en la Figura 6-2 donde en la parte superior se encuentra la información almacenada en la base de datos y en la parte inferior la copiada en el reporte médico.



### Plataforma E-Health

Nombre de paciente: JULIETA PALACIOS  
 RUN: 10501160-1  
 Edad: 49  
 Servicio: CONSULTORIO  
 Procedencia: SALAMANCA  
 Medico a cargo: Gloria Campos

6/12/2017  
 15:23:27

Examen 2: Temperatura

Temperatura: 36.37 [°C]

Figura 6-2 Resultado plataforma E-Health.

Cabe destacar que la diferencia de medición no es tan elevada y está directamente relacionada en la posición de cada sensor y en la calidad con la que están contruidos, ya que, pese a la antigüedad del termómetro de mercurio, sigue siendo el más exacto del mercado.

### 6.1.2 Medición de pulso y oxígeno en la sangre

En este caso el consultorio cuenta con un aparato que permite medir el pulso y oxígeno en la sangre, además de la presión arterial. En la Figura 6-3 se muestra la medición tomada con el aparato del consultorio y los resultados correspondientes a 72 [bpm ] en pulso y 98 [SPO2] de saturación.





Figura 6-3 Pulso y saturación, medida con sensor del consultorio.

La Figura 6-4 muestra los resultados obtenidos de pulso y saturación de oxígeno en la sangre, tal como se puede ver el primero es de 72 [bpm] y el segundo es de 96 [SPO2]. Contrastando las mediciones se observa que el pulso es el mismo y el oxígeno medido varía levemente, lo que es atribuible a la confección de cada uno de ellos.



Plataforma E-Health

Nombre de paciente: JULIETA PALACIOS  
 RUN: 10501160-1  
 Edad: 49  
 Servicio: CONSULTORIO  
 Procedencia: SALAMANCA  
 Medico a cargo: Gloria Campos

6/12/2017  
 15:23:27

Examen 4: Pulso y Saturacion

Pulso: 72 [bpm]  
 Saturacion: 96 [SPo2]

Figura 6-4 Pulso y oxígeno en la sangre, medido con plataforma E-Health.

### 6.1.3 Medición de presión arterial

En este caso la máquina del consultorio (ver Figura 6-5) muestra un valor de 126 [mmHg] para la presión diastólica y uno de 73 [mmHg] para la presión sistólica.



Figura 6-5 Presión arterial medida con aparato del consultorio.

Como contraparte se tiene el aparato que utiliza la plataforma E-Health para medir la presión arterial. Éste se puede visualizar en la Figura 6-6, también se observan los valores almacenados en la base de datos y guardados en el PDF. Sus valores son de 135 [mmHg] para la presión diastólica y de 80 [mmHg] para la sistólica. Comparándolos con los medidos por la otra máquina, varían considerablemente, pero esto no se atribuye a una falla del sensor de la plataforma E-Health, ya que el paramédico explica que, al hacer dos mediciones consecutivas de presión, los valores se elevan por haber sometido el cuerpo a una situación de estrés.



Figura 6-6 Medición de presión con plataforma E-Health.

### 6.1.4 Medición de glucosa

En este caso los dos aparatos utilizados son muy similares, la gran diferencia es que el de la plataforma E-Health permite la obtención de datos mediante un puerto Jack. En la Figura 6-7 se muestra la medición realizada con el glucómetro del consultorio el cual arroja un valor de 102 [mg/dL].

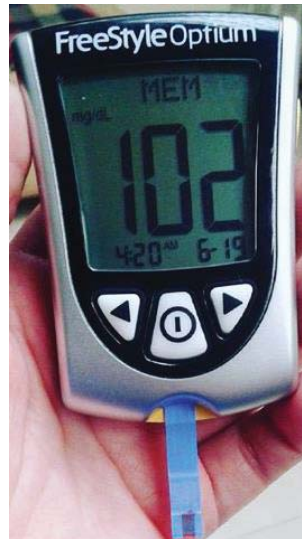


Figura 6-7 Medición de glucosa en la sangre con glucómetro estándar.

En la Figura 6-8 se muestra el valor de glucosa en la sangre obtenido por el sensor de la plataforma el cual muestra un valor de 104 [mg/dL]. Si bien las mediciones varían un poco, esto es netamente por la diferencia que existe entre los dispositivos.

**Plataforma E-Health**

**Resultados de Búsqueda**

Nombre Paciente: JULIETA PALACIOS  
 Nombre Doctor: Gloria Campos  
 Presión Diastólica: 135[mmHg]  
 Presión Sistólica: 80[mmHg]  
 Temperatura: 36.37[°C]  
 Glucosa: 104 [mg/dL]  
 Saturación: 96[SpO2]  
 Pulso: 72[mmHg]

Observaciones: Estado normal, a considerar presion un poco alta.

VOLVER

**Plataforma E-Health**

Nombre de paciente: JULIETA PALACIOS  
 RUN: 10501160-1  
 Edad: 49  
 Servicio: CONSULTORIO  
 Procedencia: SALAMANCA  
 Medico a cargo: Gloria Campos  
 Examen 3: Glucosa

Glucosa: 104 [mg/dL]

6/12/2017  
15:23:27

Figura 6-8 Medición de glucosa con plataforma E-Health.

### 6.1.5 Realización del electrocardiograma

En este caso el consultorio cuenta con un electrocardiograma muy complejo el cual se muestra en la Figura 6-9 y que tiene alrededor de 6 conexiones al cuerpo que son a torso descubierto, luego de que el paciente se encuentra con todos los electrodos puestos se realiza el examen y se va imprimiendo el diagnostico en un papel milimetrado, el cual será analizado posteriormente por el médico tratante. Dicho reporte se muestra en la Figura 6-10 y posee 6 fases.

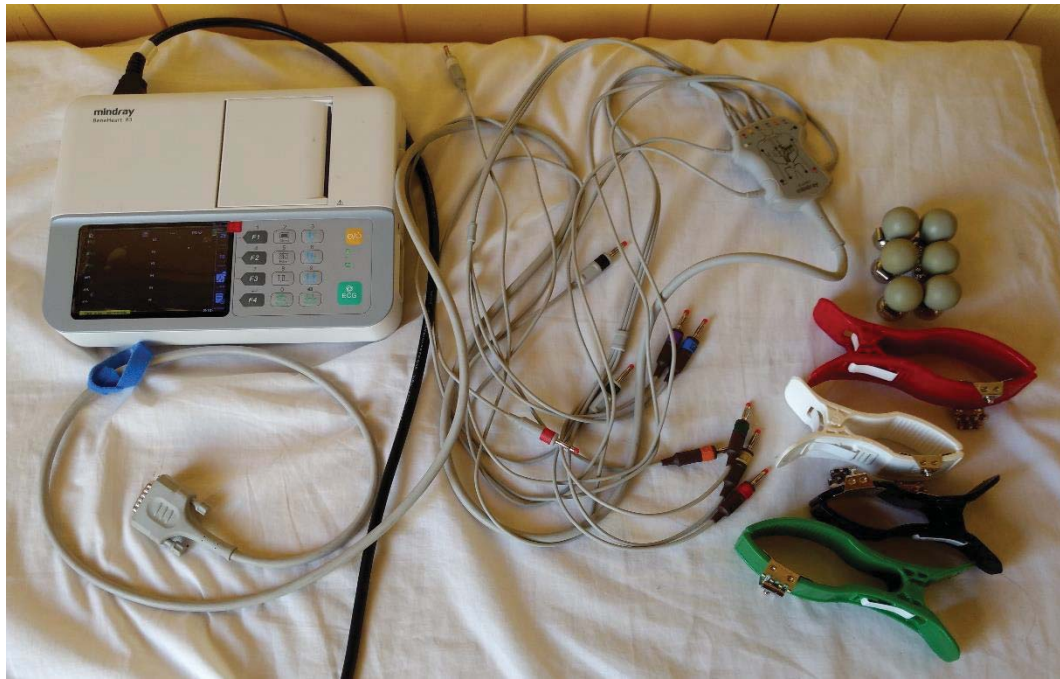


Figura 6-9 Electrocardiograma del consultorio.

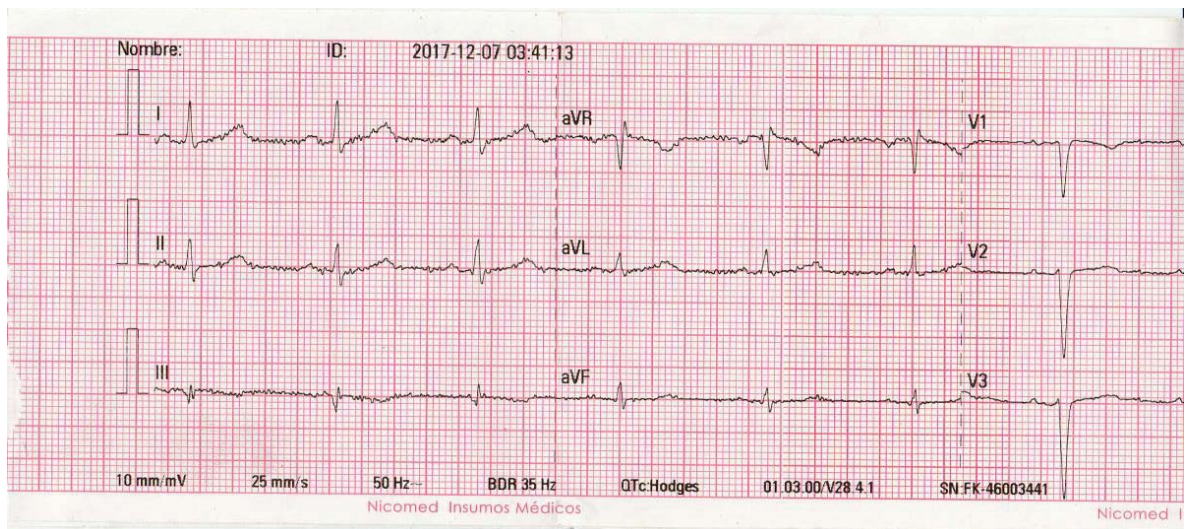


Figura 6-10 Parte del electrocardiograma tomado por la máquina del consultorio.

A diferencia del equipo explicado anteriormente, la plataforma E-Health cuenta con tres electrodos los cuales son desechables y muy fáciles de instalar, en este caso solo se muestra una fase solamente (ver Figura 6-11).

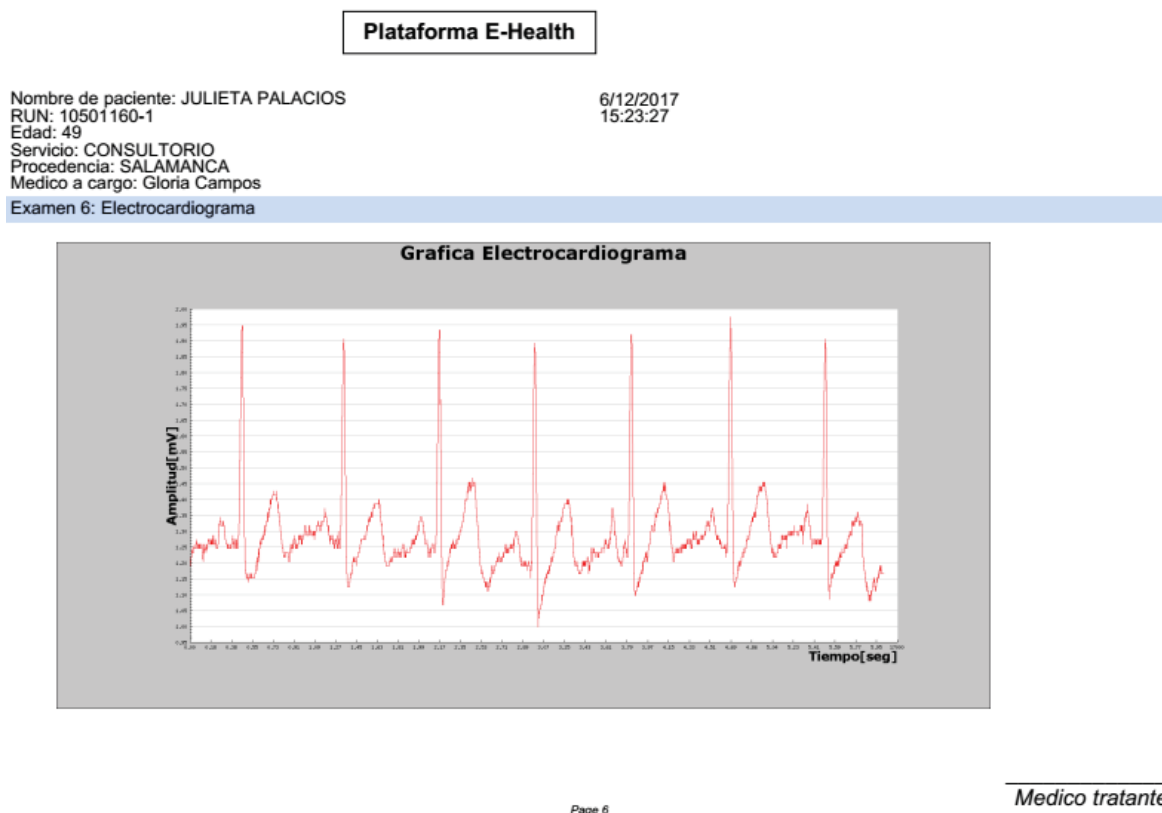


Figura 6-11 Electrocardiograma entregado por la plataforma E-Health.

Por lo tanto, pese a que en el dispositivo portátil el examen es más sencillo, la paramédico puede dar un diagnóstico importante a partir de él y explicar que se puede apreciar con facilidad el comportamiento de los pulsos y la ocurrencia de los procesos sístole y diástole.

### 6.1.6 Medición de flujo respiratorio

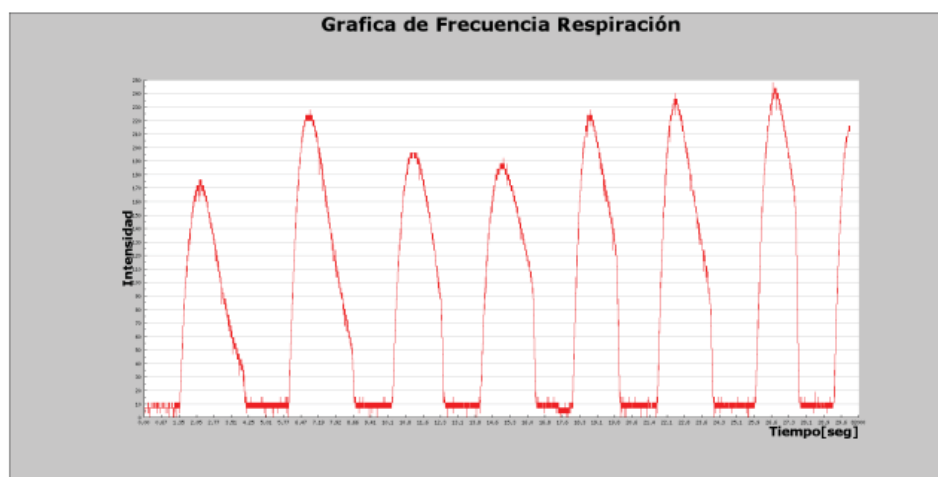
En este caso el consultorio no cuenta con un dispositivo que permita el análisis del flujo respiratorio que desarrolla una persona. Pero para este examen la paramédico ayudó a analizar los datos. La paciente estudiada Julieta Palacios tiene asma crónica por lo que su respiración es menos intensa y más lenta de lo normal. Se registraron 7 peaks (ver Figura 6-12) de expiración y lo normal deberían ser de 9 a 10 considerando que una persona respira alrededor de 20 veces en un minuto. Cabe mencionar que el personal médico de este consultorio expuso que contar con uno de estos sensores es muy importante, ya que permite analizar y realizar un seguimiento de pacientes con problemas respiratorios crónicos como Julieta.

Plataforma E-Health

Nombre de paciente: JULIETA PALACIOS  
 RUN: 10501160-1  
 Edad: 49  
 Servicio: CONSULTORIO  
 Procedencia: SALAMANCA  
 Medico a cargo: Gloria Campos

6/12/2017  
 15:23:27

Examen 5: Respiracion



Page 5

*Medico tratante*

Figura 6-12 Medición flujo respiratorio con plataforma E-Health.

### 6.1.7 Medición de un electromiograma

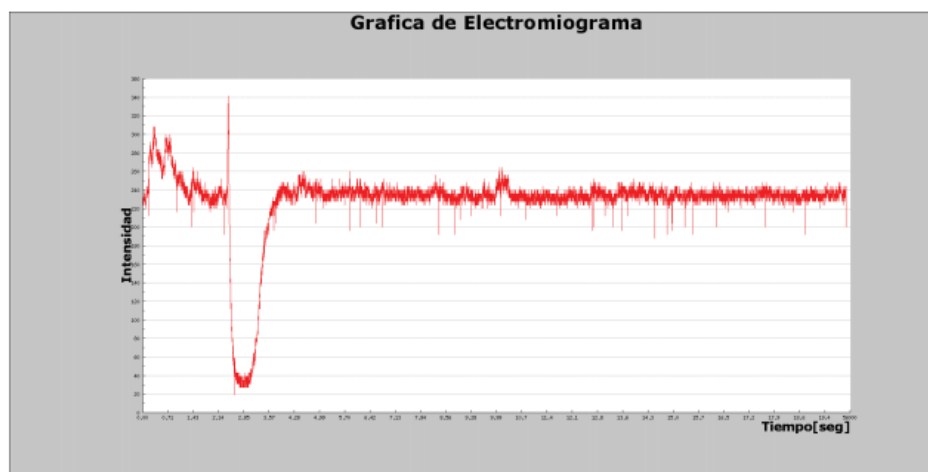
Para este examen el consultorio tampoco contaba con un electromiograma, las razones se remontan principalmente a que es un examen muy específico ya que está enfocado a diagnosticar con precisión enfermedades neuromusculares congénitas o adquiridas, y permite clasificarlas según su intensidad y origen. Esto se registra mediante los electrodos que se adhieren en los nervios y músculos afectados, con la finalidad de ver el comportamiento de las corrientes eléctricas musculares.

La Figura 6-13 muestra el electromiograma realizado a la paciente. Ella no presentaba ningún tipo de malestar muscular por lo que la prueba realizada se hizo en el músculo bíceps y se le solicitó que lo tensionara fuerte por un momento y luego después de un relajo lo mantuviera tenso hasta finalizado el examen. Y es exactamente lo que se observa en el gráfico mostrado.

Plataforma E-Health

Nombre de paciente: JULIETA PALACIOS  
 RUN: 10501160-1  
 Edad: 49  
 Servicio: CONSULTORIO  
 Procedencia: SALAMANCA  
 Medico a cargo: Gloria Campos  
 Examen 7: Electromiograma

6/12/2017  
 15:23:27



Page 7

Medico tratante

Figura 6-13 Electromiograma realizado.

La paramédico comenta que en zonas rurales como Salamanca donde la principal actividad económica es la minería o agricultura, sería de gran utilidad contar con un dispositivo que realizara este tipo de exámenes, ya que las lesiones musculares son muy frecuentes por parte de los trabajadores y temporeros.

## 6.2 Cálculo de errores en diferentes mediciones

En esta sección se calculará el error de cada una de las mediciones tomadas a 5 pacientes, contrastando los valores de la plataforma E-Health con los sensores biométricos del consultorio. En este caso sólo se comparan los exámenes que entregan valores reales y no requieren de una interpretación gráfica.

### 6.2.1 Tabla de resultados

En la Tabla 6-1 se adjuntan todos los valores medidos a 5 pacientes, cada uno de ellos posee una medición hecha con los sensores biométricos del consultorio y otra con la plataforma E-Health. Se puede apreciar que los errores más considerables se dan en la toma de la presión arterial, esto no quiere decir que el sensor del dispositivo portátil este malo, ya que quizás posea un pequeño error, este se acrecienta porque el paciente al ser sometido a una medición previa de presión, queda con un leve estrés que genera valores más elevados. Sin embargo, los errores son bastante aceptables, por lo que no se requeriría ningún tipo de calibración para estos sensores.

Tabla 6-1 Comparación resultados mediciones biométricas.

Paciente	Variable a medir	Maquina Consultorio	Plataforma E-Health	Error
Julieta Palacios	Presión Diastólica [mmHg]	126	135	4,9
	Presión Sistólica [mmHg]	76	85	5,0
	Glucosa [mm/dL]	102	104	2,0
	Temperatura [°C]	36,8	36,37	1,2
	Pulso [bpm]	72	72	0,0
	Saturación de oxígeno [SPO2]	98	96	2,0
Alexander Pérez	Presión Diastólica [mmHg]	100	105	5,0
	Presión Sistólica [mmHg]	70	73	4,2
	Glucosa [mm/dL]	102	103	1,0
	Temperatura [°C]	36,2	36,83	1,7
	Pulso [bpm]	72	70	2,8
	Saturación de oxígeno [SPO2]	99	98	1,0
Grace Rojas	Presión Diastólica [mmHg]	110	115	4,5
	Presión Sistólica [mmHg]	75	78	3,8
	Glucosa [mm/dL]	89	90	1,1
	Temperatura [°C]	36,5	36,89	1,1
	Pulso [bpm]	65	66	1,5
	Saturación de oxígeno [SPO2]	98	99	1,0
Miguel Palacios	Presión Diastólica [mmHg]	115	120	4,3
	Presión Sistólica [mmHg]	71	74	4,2
	Glucosa [mm/dL]	80	79	1,3



	Temperatura [°C]	36,1	36,61	1,4
	Pulso [bpm]	61	62	1,6
	Saturación de oxígeno [SPO2]	99	99	0,0
Mabel Palacios	Presión Diastólica [mmHg]	120	124	3,3
	Presión Sistólica [mmHg]	82	86	4,9
	Glucosa [mm/dL]	110	110	0,0
	Temperatura [°C]	36,4	36,91	1,4
	Pulso [bpm]	76	77	1,3
	Saturación de oxígeno [SPO2]	97	96	1,0

Lo errores mayores se aprecian en los resultados de la presión sistólica y diastólica. Estos tienen una explicación muy razonable y que compartió la paramédico a cargo. Sucede que cuando una persona se somete al examen de presión arterial, los músculos, más bien el organismo queda estresado, provocando que al realizar una segunda medición de forma inmediata los valores se aumenten o disminuyan dependiendo de la reacción que tenga el cuerpo de la persona a este efecto. Cabe destacar que primero se realizó la medición con el sensor del consultorio y luego con el dispositivo portátil.

## Discusión y conclusiones

El sistema de salud del país es muy precario y cuenta con muchas falencias que hacen que los hospitales y consultorios vivan en un constante colapso. Actualmente los tiempos de espera para ver a un especialista son demasiado extensos, a esto se suma que Chile es un país centralizado y la gran parte de médicos con especialidad se encuentran en las grandes ciudades, impidiendo que las personas más aisladas puedan acceder a una atención de ese tipo de forma fácil y rápida

Una solución a estos inconvenientes es la incorporación de la telemedicina en el país, ya que realiza la aplicación de la medicina a distancia por medio de diferentes dispositivos que permiten la comunicación de los especialistas con personal médico de menor rango. De este modo se podrían atender a una mayor cantidad de personas de forma más rápida y eficiente, a pesar de que se encuentren en lugares remotos del territorio nacional.

Este proyecto contaba ya con antecedentes en dos proyectos anteriores, que tenían como base y fundamento crear un dispositivo portátil, enfocado en aplicar la telemedicina al sistema de salud nacional. Que contara con una amplia gama de sensores que fueran capaces de tomar mediciones de gran relevancia en zonas aisladas del país, como por ejemplo un electrocardiograma. Que permitiera enviar los datos obtenidos vía correo electrónico utilizando la red 3G, como fuente de acceso a internet.

A pesar del gran avance que se realizó en las dos iteraciones anteriores, aún faltaban muchas mejoras que realizar a la plataforma médica para llegar a un producto final. Es así como a lo largo de este proyecto, se buscó la forma más óptima, cómoda y eficiente, de llegar a un dispositivo final que permita implementar y abordar un servicio de telemedicina completo. Que además realizara un seguimiento en las mediciones tomadas a los pacientes, mediante la creación de una base de datos.

Sin embargo, la implementación de modificaciones en un sistema ya encaminado no fue una tarea sencilla, ya que se debió estudiar a fondo cada dispositivo obtenido y cada programa creado para entender de raíz el proyecto. De forma que, al momento de realizar las modificaciones y nuevas incorporaciones al hardware y software, no existieran grandes inconvenientes que retrasaran el avance del proyecto.

Primero se analizó la plataforma E-Health, que corresponde al hardware médico utilizado y desarrollado por Cooking Hacks, las consideraciones más relevantes fueron en base a los protocolos de comunicación que utilizaba para enlazarse con la Raspberry Pi, quien fue el cerebro de este proyecto. Luego de ello se investigaron las funcionalidades y los protocolos que regían la operación y comunicación del ordenador. Siendo una de los principales inconvenientes el hecho de que la plataforma solo funcionaba con el modelo 2 de la Raspberry Pi.

Para cumplir con todas las funcionalidades esperadas se tuvo que llegar a un sistema que utilizando la Raspberry Pi como cerebro permitiera la conexión de una pantalla, un módulo 3G y la plataforma E-Health, con el fin de que el dispositivo cumpliera con todas las funcionalidades exigidas. De esta forma se eligió la pantalla táctil desarrollada por la Fundación Raspberry Pi, ya que presenta una mayor estabilidad y un controlador dedicado (vía puerto DSI) tanto al procesamiento de la señal visual, como táctil. En el caso del módulo Wi-Fi, se seleccionó el Dongle Huawei, ya que es sencillo y cómodo de implementar, ya que por medio del puerto USB la Raspberry Pi se puede conectar a internet.

Se realizó un diseño general de cómo sería el dispositivo final. Luego de ello se realizaron los diseños del banco de baterías, el soporte para adherir la pantalla en la parte superior de la maleta y la caja que contendría la plataforma E-Health. Una vez diseñadas todas las piezas se imprimieron y se armó la maleta tal como se propuso, realizando un rotulado en la caja de conexión, una subdivisión de cada sensor. De esta forma se llegó a la obtención del dispositivo final que permite su traslado y utilización en cualquier zona del país.

En la segunda iteración se creó una interfaz gráfica para el sistema, esta funcionaba por medio de una pantalla táctil y estaba desarrollada en el lenguaje PHP. Elegido principalmente por la facilidad que entregaba en caso de querer llevar todo este software a un servidor web. En ella se podían obtener y visualizar todos los datos en tiempo real además de disponer de una herramienta lista para realizar el envío de datos vía mail. Sin embargo el sistema tenía un funcionamiento lento y un mal aspecto visual. En este caso era de suma relevancia realizar el cambio de estas dos variantes, mejorando la visual y rapidez de la interfaz.

Al principio los cambios realizados dejaron sin funcionamiento la obtención de datos médicos, por lo que se tuvo que crear un estilo genérico que permitiera cargar una interfaz base cada vez que fuese necesario. Lo que antes se hacía de forma manual en cada código utilizado ralentizando demasiado las acciones del sistema. En el menú principal no se pudo realizar una renovación completa más que mantener el estilo, ya que cualquier tipo de variante dejaba inoperativo el sistema. Finalmente se logró mejorar el aspecto visual y aumentar la rapidez en la toma de datos y cambio de interfaz.

Respecto a la creación de la base de datos, corresponde a una de las modificaciones más importantes dentro del software, ya que se realizaron tablas de almacenamiento de información desde cero. Estas permitían almacenar las mediciones tomadas dentro del software, con el fin de visualizarlas a lo largo del tiempo. También está diseñada para que se pueda replicar en cualquier servidor permitiendo así su visualización vía web. El hecho de generar un historial completo de

todas las veces que los pacientes van a realizarse un examen, permite diferenciar cuál es su evolución o por lo contrario el avance de la enfermedad que pueda presentar cada paciente.

Una vez que el dispositivo estuvo terminado en su totalidad, se llevó al consultorio de la localidad de Llimpo, donde la paramédico a cargo Gloria Campos, se encargó de medir y comparar todos los datos obtenidos con los sensores biomédicos que el Ministerio de salud le proporciona para examinar a sus pacientes. Los resultados obtenidos fueron bastante razonables para ser una máquina prototipo y que realiza muchas funciones a la vez, a diferencia de los sensores biométricos utilizados, donde que cada uno de ellos cumple su función específica.

Lo más relevante en esta prueba en terreno fue que la máquina cuenta con los que el consultorio no disponía, como el electromiograma y el de flujo respiratorio. También el hecho de poder visualizar de forma inmediata los resultados por medio de la pantalla del sistema y luego de anotar las observaciones enviarlos vía mail a cualquier persona. Causaron un gran interés por parte del personal médico, ya que expuso que se trata de una idea muy buena e innovadora, que está a años de aplicarse en concreto en nuestro país ya que no existen personas que planteen esos proyectos. Una de las cosas a la que mayor importancia le dio fue la disposición de una base de datos, ya que en centros como esos se controla mucho a pacientes que poseen enfermedades crónicas como diabetes y asma.

Finalmente se da por finalizado este proyecto, ya que se logró el objetivo que era la realización de un dispositivo médico final. Otro hecho relevante es que la placa utilizada ya se encuentra obsoleta, sin soporte y fuera de producción por lo que si se quisiera seguir iterando este proyecto, lo mejor sería utilizar una plataforma más moderna, que tenga un soporte amplio y que se siga produciendo por unos años más. En caso de querer replantear este proyecto y modernizarlo, sería de suma importancia que se realizara de una sola vez y no en varias iteraciones como lo fue este caso, ya que la tecnología avanza muy rápido y es muy difícil entender y editar el trabajo ya realizado por otra persona.

# Bibliografía

- [1] V. Viñals, «Diario y Radio U de Chile,» Diario U de Chile, 14 2 2017. [En línea]. Available: <http://radio.uchile.cl/2017/02/14/demandan-reforma-al-examen-de-conocimientos-en-medicina/>. [Último acceso: 21 4 2017].
- [2] Clinic Cloud, «Tipos de telemedicina,» [En línea]. Available: <https://clinic-cloud.com/ques-telemedicina-definicion-tipos/>. [Último acceso: 20 04 2017].
- [3] «Sistema de telemedicina,» [En línea]. Available: <http://www.gamp.org.br/tele-espirometria/>. [Último acceso: 20 04 2017].
- [4] C. R. Ibañez, «Telemedicina: Introducción, aplicación y principios de desarrollo,» *Ces Medicina*, vol. 21, n° 1, pp. 77-93, 2007.
- [5] K. Nader, «El Hospital,» [En línea]. Available: <http://www.elhospital.com/temas/Paises-con-telemedicina-avanzada+8088177?pagina=1>. [Último acceso: 20 04 2017].
- [6] Accuhealth, «Telemedicina en Chile,» [En línea]. Available: <https://www.accuhealth.cl/convenios/cruzblanca-2/>. [Último acceso: 20 04 2017].
- [7] F. Siebert, «Teniendo puentes hacia una mejor salud,» Universidad de Chile, 06 07 2016. [En línea]. Available: <http://www.uchile.cl/noticias/123326/proyecto-echo-una-nueva-propuesta-de-telemedicina-para-el-hospital>. [Último acceso: 20 04 2017].
- [8] Cardiovox, «ECG PORTATIL con Bluetooth,» [En línea]. Available: <http://cardiovox.tiendas.coop/ecg-portatil-conbluetooth>. [Último acceso: 22 04 2017].
- [9] Libelium, «MySignals SW BLE Complete Kit (eHealth Medical Development Platform,» The IoT Market Place, [En línea]. Available: <https://www.the-iot-marketplace.com/mysignals-sw-ehealth-medical-biometric-complete-kit-ble>. [Último acceso: 23 04 2017].

- 
- [10] Raspberry Pi Foundation, «Raspberry Pi Modelo 2,» [En línea]. Available: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. [Último acceso: 23 04 2017].
- [11] Cooking Hacks, «e-Health Sensor Platform V2.0 [Biometric / Medical Applications],» [En línea]. Available: <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical/>. [Último acceso: 24 04 2017].
- [12] DIYMakers, «DIYMakers,» 7 8 2014. [En línea]. Available: <http://diymakers.es/raspberry-pi-como-servidor-web/>. [Último acceso: 21 4 2017].
- [13] Olimex, «Pantalla Tactil 7",» [En línea]. Available: <http://www.mcielectronics.cl/shop/product/pantalla-touch-7-para-raspberry-pi-11377>. [Último acceso: 22 04 2017].
- [14] Huawei, «Huawei,» [En línea]. Available: <http://consumer.huawei.com/py/mobile-broadband/dongles/features/e303-py.htm>. [Último acceso: 25 Mayo 2017].
- [15] T. Instrument, «Texas Instrument,» Noviembre 2010. [En línea]. Available: <http://www.ti.com/lit/ug/sprugp1/sprugp1.pdf>. [Último acceso: 25 Mayo 2017].
- [16] R. Murphy, «Cypress Embedded in Tomorrow,» [En línea]. Available: <http://www.cypress.com/file/134171/download>. [Último acceso: 25 Mayo 2017].
- [17] A. M. Fernández, «Universidad de Córdoba,» 2004. [En línea]. Available: <http://www.uco.es/~el1mofer/Docs/IntPerif/Bus%20I2C.pdf>. [Último acceso: 26 Mayo 2017].
- [18] S. S. Something, «Sparkfun Star Something,» [En línea]. Available: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>. [Último acceso: 25 Mayo 2017].
- [19] E. Upton, «Raspberry Pi Foundation,» 14 Julio 2014. [En línea]. Available: <https://www.raspberrypi.org/blog/introducing-raspberry-pi-model-b-plus/>. [Último acceso: 25 Mayo 2017].
- [20] N. Robinson, «Hackster, learning hardware community.,» 10 Marzo 2017. [En línea]. Available: <https://www.hackster.io/nrobinson2000/how-to-run-particle-on-raspberry-pi-headless-on-pi-zero-w-cd3ca2>. [Último acceso: 25 Mayo 2017].
- [21] Adafruit, «Pantalla Adafruit para Raspberry Pi,» [En línea]. Available: <https://www.adafruit.com/product/2260>. [Último acceso: 25 05 2017].

- 
- [22] Adafruit, «Pantalla Raspberry Pi Foundation 7",» [En línea]. Available: <https://www.adafruit.com/product/2718>. [Último acceso: 25 05 2017].
- [23] C. Hacks, «Cooking Hacks,» [En línea]. Available: <https://www.cooking-hacks.com/documentation/tutorials/3g-gps-shield-arduino-raspberry-pi-tutorial/>. [Último acceso: 25 Mayo 2017].
- [24] ECURED, «[www.ecured.cu](http://www.ecured.cu),» ECURED, 13 11 2015. [En línea]. Available: [https://www.ecured.cu/Interfaz\\_gr%C3%A1fica\\_de\\_usuario\\_\(GUI\)](https://www.ecured.cu/Interfaz_gr%C3%A1fica_de_usuario_(GUI)). [Último acceso: 15 10 2017].
- [25] C. Aubry, HTML5 y CSS3 para sitios con diseño web responsive, Barcelona: Ediciones ENI, 2014.
- [26] Apache, «The apache software foundation,» Apache, [En línea]. Available: <https://www.apache.org/foundation/>. [Último acceso: 3 12 2017].
- [27] MySQL, «MySQL,» [En línea]. Available: <https://www.mysql.com/>. [Último acceso: 3 12 2017].
- [28] phpMyAdmin, «phpMyAdmin,» [En línea]. Available: <https://www.phpmyadmin.net/>. [Último acceso: 3 12 2017].

# A Manual de uso

Recuerde que en los apéndices debe utilizar los estilos desde “Título 6” a “Título 9”, para generar sus títulos, subtítulos, etc.

A continuación, se explicará paso a paso como se debe utilizar el dispositivo portátil. Indicando como se hay encender la maleta y remover la pantalla táctil para visualizar la interfaz gráfica de manera más cómoda, donde se encuentran los sensores y como se deben conectar a la persona y a la plataforma E-Health, para asegurar que las mediciones se hagan correctamente. Por último, se entregará la ubicación del respaldo del sistema operativo completo y que se debería hacer en caso de que ocurra algún inconveniente.

## A.1 Encendido de la plataforma

Al abrir la maleta nos encontraremos con la pantalla táctil, la caja del banco de baterías, la caja contenedora de la plataforma E-Health y las subdivisiones que guardan cada uno de los sensores biométricos, tal como se muestra en la Figura A-1.



Figura A-1 Maleta final rotulada.



Primero debemos levantar la pantalla, para ello la tomamos de los costados y la movemos a la izquierda hasta que llegue al tope, luego la subimos hasta que tope nuevamente y la corremos hacia la derecha hasta que se fije. De esta forma nos aseguramos de que no caiga al utilizar el táctil de la pantalla.

Luego encendemos el sistema presionando el interruptor presente en el banco de baterías (ver Figura A-2), se deja el I como activado, de esta forma el sistema se inicia y luego de 1 minuto aparece la interfaz gráfica operativa en pantalla.



Figura A-2 Banco de baterías, con switch de encendido.

En la Figura A-1 se muestran unos números, éstos indican el orden de los 7 sensores que se guardan en la maleta y se enumeran a continuación:

1. Sensor de Pulso y Oxígeno en la sangre.
2. Sensor de Flujo Respiratorio
3. Sensor de glucosa en la sangre.
4. Sensor de presión arterial.
5. Sensor de temperatura corporal.
6. Sensor para electromiograma.
7. Sensor para electrocardiograma.

### A.1.1 Conexión sensor de pulso y oxígeno en la sangre

Este sensor se conecta en la ranura cuadrada que se encuentra en la parte superior de la caja contenedora y dice SPO2 tal como se muestra en la Figura A-3. Luego se pone en el dedo índice del paciente. Para iniciar la medición se debe encender el sensor y esperar que tome las mediciones. Una vez obtenidas, en la interfaz gráfica se presiona el botón “Pulso y oxígeno”. De esta forma se mostrarán los datos en pantalla.



Figura A-3 Conexión sensor de pulso y oxígeno.

### A.1.2 Conexión sensor de flujo respiratorio

Se conecta el sensor en la zona nasal tal como se muestra en la Figura A-4 y el conector al terminal Jack que se encuentra en la parte derecha y está bajo de FR. Luego se presiona el botón “Graficas” en el menú principal, lo cual nos re direcciona al menú gráficas. Una vez dentro se presiona el botón “Respiración” con el fin de iniciar el proceso de medición, se espera hasta que aparezca el gráfico correspondiente y se vuelve a la ventana principal.

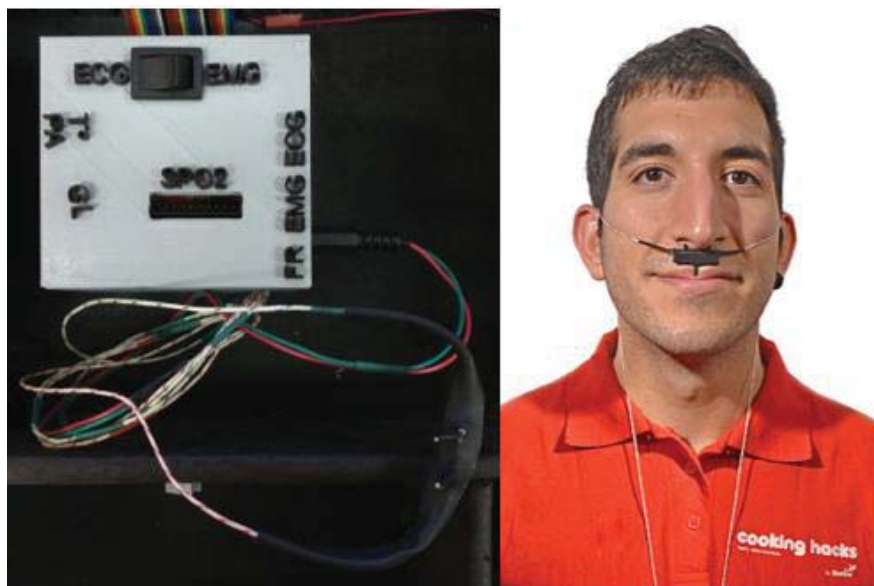


Figura A-4 Conexión sensor flujo respiratorio.

### A.1.3 Conexión sensor de glucosa

Primero se realiza la medición en el glucómetro, para ello se adhiere la tira reactiva al sensor y luego con una lanceta estéril se perfora un dedo para obtener la sangre, posteriormente se añade la sangre a la tira reactiva y se espera un par de segundos. Una vez terminada la medición y visualizados los valores en pantalla, se conecta el glucómetro al terminal Jack que se encuentra rotulado como GL al costado izquierdo de la caja contenedora (ver Figura A-5). Para iniciar el traspaso de los datos a la plataforma se presiona el botón “Glucómetro” en la interfaz gráfica. De esta forma se mostrarán los datos en pantalla.

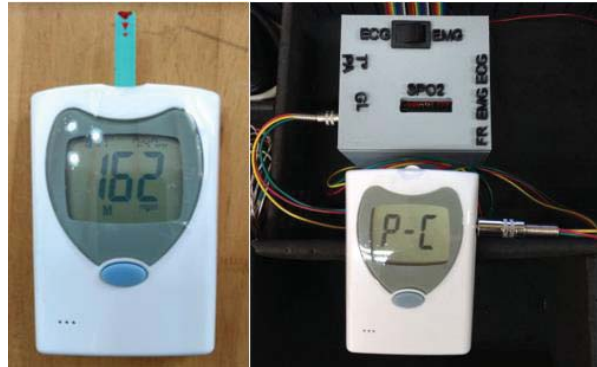


Figura A-5 Conexión sensor de glucosa.

### A.1.4 Conexión sensor de presión arterial

Primero se realiza la medición en el esfigmomanómetro, para ello se coloca en el bíceps y se presiona el botón encender. Una vez terminada la medición y visualizados los valores en pantalla, se conecta el glucómetro al terminal Jack que se encuentra rotulado como PA en la parte superior izquierda de la caja contenedora (ver Figura A-6). Para iniciar el traspaso de los datos a la plataforma se presiona el botón “Presión” en la interfaz gráfica. De esta forma se mostrarán los datos en pantalla.



Figura A-6 Conexión sensor de presión arterial.

### A.1.5 Conexión sensor de temperatura

Este sensor se conecta al terminal Jack que se encuentra rotulado como T° en el lado superior izquierdo de la caja contenedoras y en el dedo índice del paciente (ver Figura A-7). Para iniciar la medición se presiona el botón “Temperatura” en la interfaz gráfica. De esta forma se mostrarán los datos en pantalla.

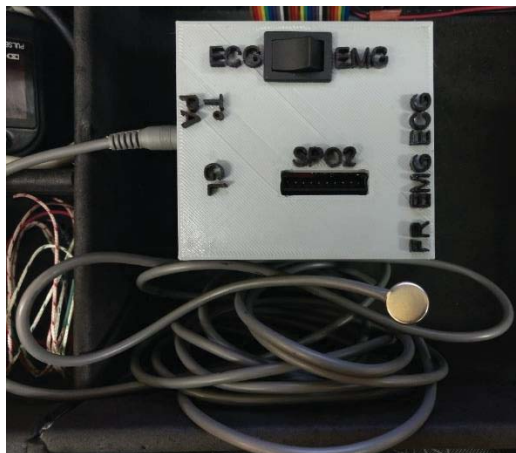


Figura A-7 Conexión sensor de temperatura.

### A.1.6 Conexión sensor para electromiograma

Se conecta el sensor en la zona bíceps y el conector al terminal Jack que se encuentra rotulado como EMG en la parte derecha de la caja contenedora y el interruptor debe estar abajo donde dice EMG (ver Figura A-8). El color rojo corresponde a la señal M, el color azul a la señal E y el color negro a GND. Luego se presiona el botón “Gráficas” en el menú principal, lo cual nos re direcciona al menú gráficas. Una vez dentro de él se presiona el botón “Electromiograma” con el fin de iniciar el proceso de medición, se espera hasta que aparezca el gráfico correspondiente y se vuelve a la ventana principal.

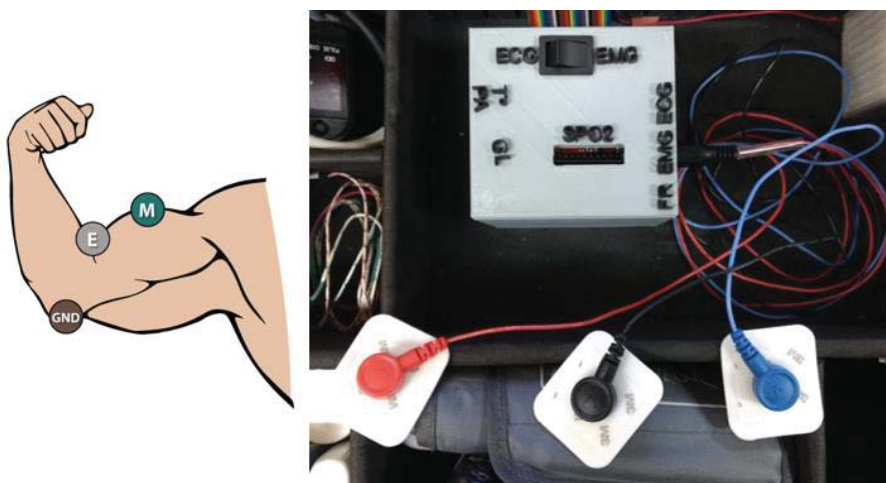


Figura A-8 Conexión sensor para electromiograma.

### A.1.7 Conexión sensor para electrocardiograma

Se conecta el sensor en el tronco de la persona y el conector al terminal Jack que se encuentra rotulado como ECG en la parte derecha de la caja contenedora y el interruptor debe estar abajo donde dice ECG (ver Figura A-9). El color rojo corresponde a la señal +, el color azul a la señal N y el color negro a - Luego se presiona el botón “Gráficas” en el menú principal, lo cual nos re direcciona al menú gráficas. Una vez dentro de él se presiona el botón “Electromiograma” con el fin de iniciar el proceso de medición, se espera hasta que aparezca el grafico correspondiente y se vuelve a la ventana principal.

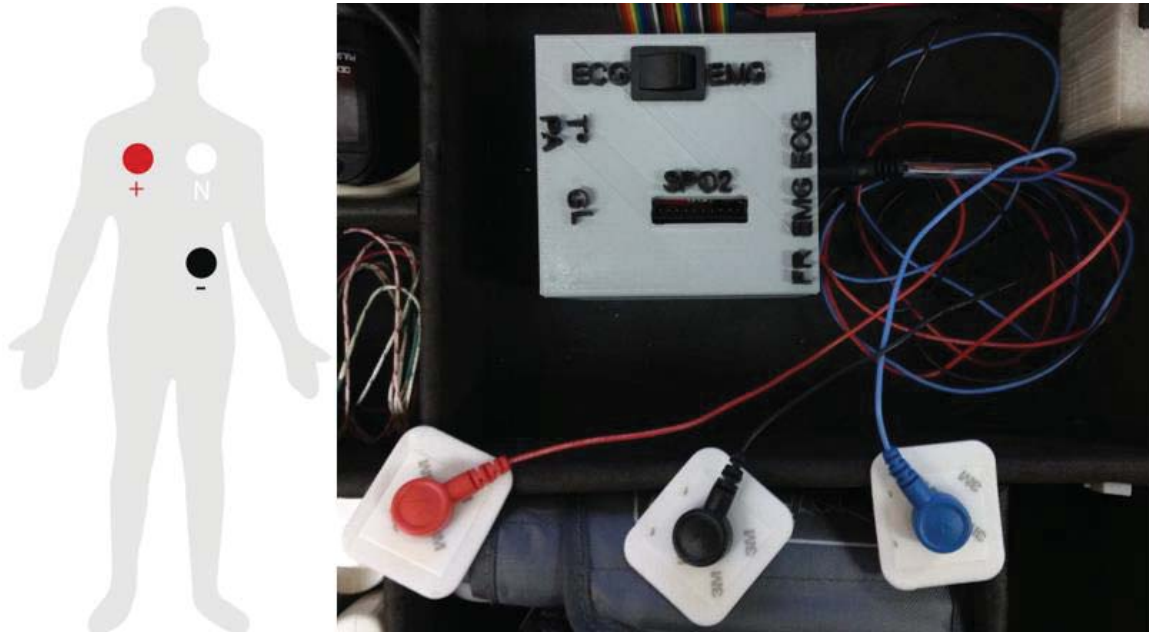


Figura A-9 Conexión sensor para electrocardiograma.

## A.2 Como solucionar inconvenientes

En esta sección se muestra y adjunta material que nos permite solucionar inconvenientes en caso de que la maquina falle, principalmente las fallas pueden presentarse en el área de la programación, ya que al contar con muchos lenguajes que conforman este sistema, muchas veces las mismas actualizaciones de programas dejan inoperativas algunas funciones.

### A.2.1 Caso de perdida en la lectura de los sensores

Esto lo podemos solucionar desarmando el sistema de soporte de pantalla con un destornillador y mucho cuidado, luego retiramos la parte posterior que protege a la Raspberry Pi y tiene los enganches correspondientes para poder elevar la pantalla. De esta forma podremos conectar un teclado y mouse que nos permita acceder a los programas.

Una vez desmontado el soporte y conectados los dispositivos externos, presionamos Alt+F4 para cerrar la ventana de la interfaz gráfica y acceder al escritorio del sistema operativo. En seguida nos

dirigimos al sistema de ficheros, específicamente a la dirección /var/www, que es donde se encuentran todos los programas, luego buscamos las versiones CPP y PHP del sensor que está fallando y se comparan con lo que se encuentran en el respaldo online ( <https://www.dropbox.com/sh/8pr014i9oellmop/AACqGUbqfrOnAtYxkoRfw8vga?dl=0> ).

### **A.2.2 Caso de falla en todo el sistema**

En caso de que falle todo el sistema, tanto la base de datos, las lecturas de los sensores o la propia interfaz, debemos realizar el mismo proceso de desmontaje mencionado anteriormente, pero extraer la memoria SD que posee la Raspberry Pi.

Una vez que disponemos de ella, debemos formatear en nuestro ordenador la tarjeta de memoria, para ello se puede utilizar el programa SD formater que la reestablece desde fabrica. Luego que esta formateada cargamos la imagen del sistema que está en el link de Dropbox <https://www.dropbox.com/sh/8pr014i9oellmop/AACqGUbqfrOnAtYxkoRfw8vga?dl=0> . esto se puede hacer mediante el programa Win32DiskImager, que permite copiar toda la imagen que contiene el sistema operativo y recuperar la funcionalidad del dispositivo portátil.

# B Programación dispositivo portátil

El dispositivo requiere de la mezcla de varios lenguajes de programación para lograr la obtención de las mediciones de cada sensor y permitir su visualización por parte del usuario. Por ello se muestran a continuación los códigos de programación de cada sensor, además de la librería usada, la cual se puede modificar con el fin de calibrar los sensores que tomen mediciones incorrectas. También se exponen los códigos de la interfaz gráfica que permite mostrar de forma inmediata los resultados de los exámenes.

## B.1 Programación sensores biométricos

En esta sección se muestran los códigos correspondientes a los sensores de temperatura, pulso y oxígeno, presión, glicemia, flujo respiratorio, electromiograma, electrocardiograma. Además, la librería que se debe modificar en caso de una calibración requerida.

### B.1.1 Librería de programación (eHealth.cpp)

Este es el código principal de la plataforma, el cual posee las funciones para el uso de los sensores biométricos.

```
#include "eHealth.h"

//*****
// Accelerometer Variables and definitions *
//*****

    //! Breakout board defaults to 1, set to 0 if SA0 jumper is set
#define SA0 1
    #if SA0
        #define MMA8452_ADDRESS 0x1D  //! SA0 is high, 0x1C if low
    #else
        #define MMA8452_ADDRESS 0x1C
    #endif

#define int1Pin 2
#define int2Pin 3
```

```

    //! Set the scale below either 2, 4 or 8.
    const byte scale = 2;

    //! Set the output data rate below. Value should be between 0 and 7.
    //! 0=800Hz, 1=400, 2=200, 3=100, 4=50, 5=12.5, 6=6.25, 7=1.56
    const byte dataRate = 0;

    /*******
    // Constructor of the class *
    /*******

    //! Function that handles the creation and setup of instances
    eHealthClass::eHealthClass(void) { /*void constructor*/ }

    /*******
    // Public Methods *
    /*******

    /**!*****
    //! Name: initPositionSensor() *
    //! Description: Initializes position sensor configure some value.*
    //! Param : void *
    //! Returns: void *
    //! Example: eHealth.initPositionSensor(); *
    /**!*****

    void eHealthClass::initPositionSensor(void)
    {
        Wire.begin();
        byte c;
        /* Set up the interrupt pins, they're set as active high, push-pull */
        pinMode(int1Pin, INPUT);
        digitalWrite(int1Pin, LOW);
        pinMode(int2Pin, INPUT);
        digitalWrite(int2Pin, LOW);
        /* Read the WHO_AM_I register, this is a good test of communication */
        c = readRegister(0x0D); // Read WHO_AM_I register
        if (c == 0x2A) { // WHO_AM_I should always be 0x2A
            initMMA8452(scale, dataRate);
            printf("MMA8452Q is online...\n");
        } else {
            printf("Could not connect to MMA8452Q: 0x%X\n",c);
            //while (1);
        }
    }
}

```



```

//!*****
//!  Name: readBloodPressureSensor()                               *
//!  Description: Initializes the BloodPressureSensor sensor.    *
//!  Param : void                                                *
//!  Returns: void                                               *
//!  Example: eHealth.initBloodPressureSensor();                 *
//!*****

void eHealthClass::readBloodPressureSensor(void)
{
    unsigned char _data;
    int ia=0;
    length=0;

    Serial.begin(19200);
    Serial.write(0xAA);
    delayMicroseconds(1);
    Serial.write(0x55);
    delayMicroseconds(1);
    Serial.write(0x88);
    delay(2500);
    Serial.print("\n");

    if (Serial.available() > 0) { // Protocol sends the measures

        for (int i = 0; i<4; i++){ // Read four dummy data
            Serial.read();
        }

        while(_data != 0xD1){
            if (ia==0){
                _data = Serial.read();
            }

            bloodPressureDataVector[length].year = swap(_data);
            bloodPressureDataVector[length].month = swap(Serial.read());
            bloodPressureDataVector[length].day = swap(Serial.read());
            bloodPressureDataVector[length].hour = swap(Serial.read());
            bloodPressureDataVector[length].minutes = swap(Serial.read());
            bloodPressureDataVector[length].systolic = swap(Serial.read());
            bloodPressureDataVector[length].diastolic = swap(Serial.read());
            bloodPressureDataVector[length].pulse = swap(Serial.read());
            length++;
            ia=1;
        }
        for (int i = 0; i<4; i++){ // CheckSum 1
            Serial.read();
        }
    }
}

```

```

        _data = Serial.read();
    }
    for (int i = 0; i<3; i++){ // CheckSum 2
        Serial.read();
    }
}
}

//!*****
//!      Name: initPulsioximeter()                               *
//!      Description: Initializes the pulsioximeter sensor.      *
//!      Param : void                                           *
//!      Returns: void                                           *
//!      Example: eHealth.initPulsioximeter();                   *
//!*****

void eHealthClass::initPulsioximeter(void)
{
    // Configuring digital pins like INPUTS
    pinMode(13, INPUT);          pinMode(12, INPUT);
    pinMode(11, INPUT);          pinMode(10, INPUT);
    pinMode( 9, INPUT);          pinMode( 8, INPUT);
    pinMode( 7, INPUT);          pinMode( 6, INPUT);
    // attach a PinChange Interrupt to our pin on the rising edge
}

//!*****
//!      Name: getTemperature()                                   *
//!      Description: Returns the corporal temperature.          *
//!      Param : void                                           *
//!      Returns: float with the corporal temperature value.    *
//!      Example: float temperature=eHealth.getTemperature();   *
//!*****

float eHealthClass::getTemperature(void)
{
    //Local variables
    float Temperature; //Corporal Temperature
    float Temperatura;
    float Resistance; //Resistance of sensor.
    float ganancia=5.0;
    float Vcc=4.1389;
    float RefTension=2.98;
    float Ra=4650.0; //Wheatstone bridge resistance.
    float Rc=4720.0; //Wheatstone bridge resistance.
    float Rb=821.0; //Wheatstone bridge resistance.
}

```



```

int eHealthClass::getOxygenSaturation(void)
{
    return SPO2;
}

//!*****
//!      Name: getBPM()                                *
//!      Description: Returns the heart beats per minute. *
//!      Param : void                                    *
//!      Returns: int with the beats per minute          *
//!      Example: int BPM = eHealth.getBPM();           *
//!*****

int eHealthClass::getBPM(void)
{
    return BPM;
}

//!*****
//!      Name: getSkinConductance()                      *
//!      Description: Returns the value of skin conductance. *
//!      Param : void                                    *
//!      Returns: float with the value of skin conductance *
//!      Example: float conductance = eHealth.getSkinConductance(); *
//!*****

float eHealthClass::getSkinConductance(void)
{
    // Local variable declaration.
    float resistance;
    float conductance;
    delay(1);

    // Read an analogic value from analogic2 pin.
    float sensorValue = analogRead(2);
    float voltage = sensorValue*5.0/1023;

    conductance = 2*((voltage - 0.5) / 100000);

    // Conductance calculation
    resistance = 1 / conductance;
    conductance = conductance * 1000000;
    delay(1);

    if (conductance > 1.0) return conductance;
    else return -1.0;
}

```

```

//!*****
//! Name: getSkinResistance() *
//! Description: Returns the value of skin resistance. *
//! Param : void *
//! Returns: float with the value of skin resistance *
//! Example: float resistance = eHealth.getSkinResistance(); *
//!*****

```

```

float eHealthClass::getSkinResistance(void)
{
    // Local variable declaration.
    float resistance;
    float conductance;

    // Read an analogic value from analogic2 pin.
    float sensorValue = analogRead(2);
    float voltage = (sensorValue * 5.0) / 1023;

    delay(2);
    conductance = 2*((voltage - 0.5) / 100000);

    //Conductance calculacion
    resistance = 1 / conductance;
    delay(2);

    if (resistance > 1.0 ) return resistance;
    else return -1.0;
}

```

```

//!*****
//! Name: getSkinConductanceVoltage() *
//! Description: Returns the skin conductance value in voltage. *
//! Param : void *
//! Returns: float with the skin conductance value in voltage *
//! Example: float volt = eHealth.getSkinConductanceVoltage(); *
//!*****

```

```

float eHealthClass::getSkinConductanceVoltage(void)
{
    delay(2);

    //Read analogic value from analogic2 pin.
    int sensorValue = analogRead(2);

    //Convert the readed value to voltage.
    float voltage = ( sensorValue * 5.0 ) / 1023;
}

```

```

        delay(2);
        return voltage;
    }

    //!*****
    //! Name: getECG() *
    //! Description: Returns an analogic value to represent the ECG. *
    //! Param : void *
    //! Returns: float with the ECG value in voltage *
    //! Example: float volt = eHealth.getECG(); *
    //!*****

    float eHealthClass::getECG(void)
    {
        float analog0;
        // Read from analogic in.
        analog0=analogRead(0);
        // binary to voltage conversion
        return analog0 = ((float)analog0 * 3.5) / 1023.0;
    }

    //!*****
    //! Name: getEMG() *
    //! Description: Returns an analogic value to represent the EMG.*
    //! Param : void *
    //! Returns: float with the EMG value in voltage *
    //! Example: float volt = eHealth.getEMG(); *
    //!*****

    int eHealthClass::getEMG(void)
    {
        int analog0;
        // Read from analogic in.
        analog0=analogRead(0);
        // binary to voltage conversion
        return analog0;
    }

    //!*****
    //! Name: getBodyPosition() *
    //! Description: Returns the current body position. *
    //! Param : void *
    //! Returns: uint8_t with the the position of the patient. *
    //! Example: uint8_t position = eHealth.getBodyPosition(); *
    //!*****

```

```

uint8_t eHealthClass::getBodyPosition(void)
{
    static byte source;

    /* If int1 goes high, all data registers have new data */
    if (digitalRead(int1Pin)) { // Interrupt pin, should probably
        readRegisters(0x01, 6, &data[0]); // Read the six data

        /* For loop to calculate 12-bit ADC and g value for each axis */
        for (int i=0; i<6; i+=2) {
            accelCount[i/2] = ((data[i] << 8) | data[i+1]) >> 4;

            if (data[i] > 0x7F) {
                accelCount[i/2] = ~accelCount[i/2] + 1;
                accelCount[i/2] *= -1;
            }

            accel[i/2] = (float) accelCount[i/2]/((1<<12)/(2*scale));
        }

        /* If int2 goes high, either p/l has changed or there's been a
        single/double tap */
        if (digitalRead(int2Pin)) {
            source = readRegister(0x0C); // Read the interrupt source reg.

            if ((source & 0x10)==0x10) // If the p/l bit is set.
                portraitLandscapeHandler();

            delay(50); // Delay here for a little printing visibility.
        }

        delay(100);

        return bodyPos;
    }

    /*!*****
    /*! Name: getSystolicPressure() *
    /*! Description: Returns the value of the systolic pressure. *
    /*! Param : void *
    /*! Returns: int with the systolic pressure. *
    /*! Example: int systolic = eHealth.getSystolicPressure(); *
    /*!*****

    int eHealthClass::getSystolicPressure(int i)
    {

```

```

        return bloodPressureDataVector[i].systolic;
    }

    //!*****
    //! Name: getDiastolicPressure() *
    //! Description: Returns the value of the diastolic pressure. *
    //! Param : void *
    //! Returns: int with the diastolic pressure. *
    //! Example: int diastolic = eHealth.getDiastolicPressure(); *
    //!*****

    int eHealthClass::getDiastolicPressure(int i)
    {
        return bloodPressureDataVector[i].diastolic;
    }

    //!*****
    //! Name: getAirFlow() *
    //! Description: Returns an analogic value to represent the air flow*
    //! Param : void *
    //! Returns: int with the airFlow value (0-1023). *
    //! Example: int airFlow = eHealth.getAirFlow(); *
    //!*****

    int eHealthClass::getAirFlow(void)
    {
        int airFlow = analogRead(1);
        return airFlow;
    }

    //!*****
    //! Name: printPosition() *
    //! Description: Returns an analogic value to represent the air flow*
    //! Param : uint8_t position : the current body position. *
    //! Returns: void *
    //! Example: eHealth.printPosition(position); *
    //!*****

    void eHealthClass::printPosition( uint8_t position )
    {
        if (position == 1) {
            printf("Prone position\n");
        } else if (position == 2) {
            printf("Stand or sit position\n");
        } else if (position == 3) {
            printf("Left lateral decubitus\n");
        } else if (position == 4) {

```



```

        printf("Supine position\n");
    } else if (position == 5) {
        printf("Rigth lateral decubitus\n");
    } else {
        printf("non-defined position\n");
    }
}

//!*****
//! Name: readPulsioximeter() *
//! Description: It reads a value from pulsioximeter sensor. *
//! Param : void *
//! Returns: void *
//! Example: readPulsioximeter(); *
//!*****

void eHealthClass::readPulsioximeter(void)
{
    uint8_t digito[200];

    uint8_t A = 0;
    uint8_t B = 0;
    uint8_t C = 0;
    uint8_t D = 0;
    uint8_t E = 0;
    uint8_t F = 0;
    uint8_t G = 0;

    for (int i = 0; i<199 ; i++) { // read all the led's
        A = !digitalRead(13);
        B = !digitalRead(12);
        C = !digitalRead(11);
        D = !digitalRead(10);
        E = !digitalRead(9);
        F = !digitalRead(8);
        G = !digitalRead(7);

        digito[i] = segToNumber(A, B, C ,D ,E, F,G);
        delayMicroseconds(43); //300 microseconds
    }
    /*
    for(int i = 0; i<199; i++){
        printf("%d", digito[i]);
        //delay(1);
    }
    printf("\n");
    delay(500);

```

```

*/

if((digito[142] != 0) &&(digito[181] == 0)){ // case 2

    SPO2 = 10 * digito[142] + digito[59];

    BPM = 100 * digito[137] + 10 * digito[10] + digito[2];
}
else if((digito[136] != 0) &&(digito[62] == 0)){ // case 3

    SPO2 = 10 * digito[179] + digito[136];

    BPM = 100 * digito[127] + 10 * digito[10] + digito[2];
}

else if((digito[145] != 0) &&(digito[62] == 0)){ // case 4
    SPO2 = 10 * digito[181] + digito[142];

    BPM = 100 * digito[50] + 10 * digito[10] + digito[2];
}

else if((digito[53] != 0) &&(digito[62] == 0)){ // case 5

    SPO2 = 10 * digito[179] + digito[53];

    BPM = 100 * digito[41] + 10 * digito[10] + digito[2];
}

else if((digito[174] != 0) &&(digito[181] == 0)){ // case 6

    SPO2 = 10 * digito[179] + digito[59];

    BPM = 100 * digito[50] + 10 * digito[10] + digito[2];
}

else{ //default case

    SPO2 = 10 * digito[179] + digito[59];

    BPM = 100 * digito[50] + 10 * digito[10] + digito[2];
}
}

//!*****
//! Name: airflowWave() *

```

```

//! Description: It prints air flow wave form in the serial monitor*
//! Param : int air with the analogic value *
//! Returns: void *
//! Example: eHealth.airflowWave(); *
//!*****

void eHealthClass::airFlowWave(int air)
{
    for (int i=0; i < (air / 5) ; i ++ ) {
        printf("..");
    }

    printf("..");
    printf("\n");
    delay(25);
}

//!*****
//! Name: readGlucometer() *
//! Description: It reads the data stored in the glucometer *
//! Param : void *
//! Returns: void *
//! Example: eHealth.readGlucometer(); *
//!*****

void eHealthClass::readGlucometer(void)
{
    // Configuring digital pins like INPUTS
    pinMode(5, OUTPUT);
    digitalWrite(5, HIGH);
    delay(100);
    Serial.begin(1200);
    delay(100);

    Serial.print("U"); // Start communication command.
    delay(1000); // Wait while receiving data.

    Serial.print("\n");
    if (Serial.available() > 0) {
        length = Serial.read();// Protocol sends measures

        if (Serial.available() > 0) Serial.read();

for (int i = 0; i<length; i++) { // Protocol sends data in this order
if (Serial.available() > 0) glucoseDataVector[i].year = Serial.read();
if (Serial.available() > 0) glucoseDataVector[i].month = Serial.read();
if (Serial.available() > 0) glucoseDataVector[i].day = Serial.read();
}
}

```

```

if (Serial.available() > 0) glucoseDataVector[i].hour = Serial.read();
if (Serial.available()>0) glucoseDataVector[i].minutes = Serial.read();
if (Serial.available() > 0)Serial.read(); // Byte of separation 0x00.
if (Serial.available()>0) glucoseDataVector[i].glucose = Serial.read();
if (Serial.available()>0) glucoseDataVector[i].meridian=Serial.read();
if (Serial.available() > 0) Serial.read(); // CheckSum 1
if (Serial.available() > 0) Serial.read(); // CheckSum 2
    }
}

        digitalWrite(5, LOW);
    }

//!*****
//!Name: getGlucometerLength() *
//!Description: it returns the number of data stored in the glucometer*
//!Param : void *
//!Returns: uint8_t with length *
//!Example: int length = eHealth.getGlucometerLength(); *
//!*****

    int eHealthClass::getGlucometerLength(void)
    {
        return length;
    }

//!*****
//! Name: getBloodPressureLength() *
//! Description: it returns the number of data stored in *
//! the blood pressure sensor *
//! Param : void *
//! Returns: uint8_t with length *
//! Example: int length = eHealth.getBloodPressureLength(); *
//!*****

    int eHealthClass::getBloodPressureLength(void)
    {
        return length;
    }

//!*****
//! Name: numberToMonth() *
//! Description: Convert month variable from numeric to character. *
//! Param : int month in numerical format *
//! Returns: String with the month characters (January, February..) *
//! Example: Serial.print(eHealth.numberToMonth(month)); *
//!*****

```

```

const char *eHealthClass::numberToMonth(int month)
{
    if (month == 1) return "enero";
    else if (month == 2) return "Febrero";
    else if (month == 3) return "Marzo";
    else if (month == 4) return "Abril";
    else if (month == 5) return "Mayo";
    else if (month == 6) return "Junio";
    else if (month == 7) return "Julio";
    else if (month == 8) return "Agosto";
    else if (month == 9) return "Septiembre";
    else if (month == 10) return "Octubre";
    else if (month == 11) return "Noviembre";
    else return "Diciembre";
}

//!*****
//! Name: version() *
//! Description: It check the version of the library *
//! Param : void *
//! Returns: void *
//! Example: eHealth.version(); *
//!*****

int eHealthClass::version(void)
{
    return 2.0;
}

//*****
// Private Methods *
//*****

//! This function will read the p/l source register and
//! print what direction the sensor is now facing */

void eHealthClass::portraitLandscapeHandler()
{
    byte pl = readRegister(0x10); // Reads the PL_STATUS register

    switch((pl&0x06)>>1) // Check on the LAPO[1:0] bits
    {
        case 0:
            position[0] = 0;
            break;

        case 1:

```

```

        position[0] = 1;
        break;

        case 2:
            position[0] = 2;
            break;

        case 3:
            position[0] = 3;
            break;
    }

    if (pl&0x01) // Check the BAFRO bit
        position[1] = 0;
    else
        position[1] = 1;
    if (pl&0x40) // Check the LO bit
        position[2] = 0;
    else
        position[2] = 1;

    bodyPosition();
}

/*****

//! Initialize the MMA8452 registers.

void eHealthClass::initMMA8452(char fsr, char dataRate)
{
    MMA8452Standby(); // Must be in standby to change registers

    /* Set up the full scale range to 2, 4, or 8g. */
    if ((fsr==2)|| (fsr==4)|| (fsr==8))
        writeRegister(0x0E, fsr >> 2);
    else
        writeRegister(0x0E, 0);

    /* Setup the 3 data rate bits, from 0 to 7 */
    writeRegister(0x2A, readRegister(0x2A) & ~(0x38));

    if (dataRate <= 7)
        writeRegister(0x2A, readRegister(0x2A) | (dataRate << 3));

    /* Set up portrait/landscap registers */
    writeRegister(0x11, 0x40); // Enable P/L
    writeRegister(0x13, 0x14); // 29deg z-lock,

```

```
writeRegister(0x14, 0x84); // 45deg thresh, 14deg hyst
writeRegister(0x12, 0x05); // debounce counter at 100ms

/* Set up single and double tap */
writeRegister(0x21, 0x7F); // enable single/double
writeRegister(0x23, 0x20); // x thresh at 2g
writeRegister(0x24, 0x20); // y thresh at 2g
writeRegister(0x25, 0x08); // z thresh at .5g
writeRegister(0x26, 0x30); // 60ms time limit
writeRegister(0x27, 0x28); // 200ms between taps min
writeRegister(0x28, 0xFF); // 1.275s (max value)

/* Set up interrupt 1 and 2 */
writeRegister(0x2C, 0x02); // Active high, push-pull
writeRegister(0x2D, 0x19); // DRDY int enabled, P/L enabled
writeRegister(0x2E, 0x01); // DRDY on INT1, P/L on INT2

MMA8452Active(); // Set to active to start reading
}

/*****/

//! Sets the MMA8452 to standby mode. It must be in standby to change
most register settings.

void eHealthClass::MMA8452Standby()
{
    byte c = readRegister(0x2A);
    writeRegister(0x2A, c & ~(0x01));
}

/*****/

//! Sets the MMA8452 to active mode. Needs to be in this mode to output
data

void eHealthClass::MMA8452Active()
{
    byte c = readRegister(0x2A);
    writeRegister(0x2A, c | 0x01);
}

/*****/

//! Read i registers sequentially, starting at address into the dest
byte array.
```

```
void eHealthClass::readRegisters(char address, int i, char * dest)
{
    Wire.beginTransaction(MMA8452_ADDRESS);
    Wire.read_rs(&address, dest, i);
}

/*****/

//! Read a single byte from address and return it as a byte.

char eHealthClass::readRegister(char address)
{
    char dest[1];
    Wire.beginTransaction(MMA8452_ADDRESS);
    Wire.read_rs(&address, dest, 1);
    return dest[0];
}

/*****/

//! Writes a single byte (data) into address
void eHealthClass::writeRegister(unsigned char address, unsigned
char data)
{
    char transmit[2];
    transmit[0] = address;
    transmit[1] = data;
    Wire.write(transmit, 2);
}

/*****/

//! Assigns a value depending on body position.

void eHealthClass::bodyPosition( void )
{
    if (( position[0] == 0 ) && (position[1] == 1) && (position [2] == 0))
    {
        bodyPos = 1;
    } else if (( position[0] == 1 ) && (position[1] == 1) && (position [2]
== 0)) {
        bodyPos = 1;
    } else if (( position[0] == 3 ) && (position[1] == 1) && (position [2]
== 0)) {
        bodyPos = 1;
    } else if (( position[0] == 2 ) && (position[1] == 0) && (position [2]
== 0)) {
```



```
        bodyPos = 5;
    } else if (( position[0] == 2 ) && (position[1] == 1) && (position [2]
== 1)) {
        bodyPos = 5;
    } else if (( position[0] == 2 ) && (position[1] == 1) && (position [2]
== 0)) {
        bodyPos = 1;
    } else if (( position[0] == 0 ) && (position[1] == 1) && (position [2]
== 1)) {
        bodyPos = 2;
    } else if (( position[0] == 0 ) && (position[1] == 0) && (position [2]
== 1)) {
        bodyPos = 2;
    } else if (( position[0] == 1 ) && (position[1] == 1) && (position [2]
== 1)) {
        bodyPos = 3;
    } else if (( position[0] == 1 ) && (position[1] == 0) && (position [2]
== 1)) {
        bodyPos = 3;
    } else if (( position[0] == 3 ) && (position[1] == 0) && (position [2]
== 1)) {
        bodyPos = 3;
    } else if (( position[0] == 3 ) && (position[1] == 1) && (position [2]
== 1)) {
        bodyPos = 3;
    } else if (( position[0] == 1 ) && (position[1] == 0) && (position [2]
== 0)) {
        bodyPos = 4;
    } else if (( position[0] == 3 ) && (position[1] == 0) && (position [2]
== 0)) {
        bodyPos = 4;
    /*
    } else if (( position[0] == 3 ) && (position[1] == 0) && (position [2]
== 1)) {
        bodyPos = 5;
    } else if (( position[0] == 3 ) && (position[1] == 1) && (position [2]
== 1)) {
        bodyPos = 5;
    */
    } else if (( position[0] == 2 ) && (position[1] == 0) && (position [2]
== 1)) {
        bodyPos = 5;
    } else {
        bodyPos = 6;
    }
}
```

```

/*****/

//! Converts from 7 segments to number.

uint8_t eHealthClass::segToNumber(uint8_t A, uint8_t B, uint8_t C,
uint8_t D, uint8_t E, uint8_t F, uint8_t G )
{
    if ((A == 1) && (B == 1) && (C == 1) && (D == 0) && (E == 1) &&
(F == 1) && (G == 1)) {
        return 0;
    } else if ((A == 0) && (B == 1) && (C == 0) && (D == 0) && (E ==
1) && (F == 0) && (G == 0)) {
        return 1;
    } else if ((A == 1) && (B == 1) && (C == 0) && (D == 1) && (E ==
0) && (F == 1) && (G == 1)) {
        return 2;
    } else if ((A == 1) && (B == 1) && (C == 0) && (D == 1) && (E ==
1) && (F == 0) && (G == 1)) {
        return 3;
    } else if ((A == 0) && (B == 1) && (C == 1) && (D == 1) && (E ==
1) && (F == 0) && (G == 0)) {
        return 4;
    } else if ((A == 1) && (B == 0) && (C == 1) && (D == 1) && (E ==
1) && (F == 0) && (G == 1)) {
        return 5;
    } else if ((A == 1) && (B == 0) && (C == 1) && (D == 1) && (E ==
1) && (F == 1) && (G == 1)) {
        return 6;
    } else if ((A == 1) && (B == 1) && (C == 0) && (D == 0) && (E ==
1) && (F == 0) && (G == 0)) {
        return 7;
    } else if ((A == 1) && (B == 1) && (C == 1) && (D == 1) && (E ==
1) && (F == 1) && (G == 1)) {
        return 8;
    } else if ((A == 1) && (B == 1) && (C == 1) && (D == 1) && (E ==
1) && (F == 0) && (G == 1)) {
        return 9;
    } else {
        return 0;
    }
}

/*****/

//! Swap data for blood pressure mesure

char eHealthClass::swap(char _data)

```

```

    {
        char highBits = (_data & 0xF0) / 16;
        char lowBits = (_data & 0x0F) * 16;
        return ~(highBits + lowBits);
    }

/*****/

//*****
// Preinstantiate Objects *
//*****

    eHealthClass eHealth = eHealthClass();

```

### B.1.2 Sensor de temperatura (temperatura.cpp)

Este código permite tomar mediciones simultáneas de temperatura durante dos minutos con el fin de obtener una media y entregar el resultado final.

```

#include "eHealth.h"
#include "stdio.h"
#include "time.h"
FILE *doc;

float temperature;
int i=0;
int main (){
doc= fopen("temp.txt","w");
    while(1){
        temperature = eHealth.getTemperature() - 0.75;
        i = i + 10;
        delay(10);
        if (i > 120000){
            printf("Temperatura : %.2f °C\n", temperature);
            fprintf(doc, "%.2f", temperature);
            time_t rawtime;

            struct tm*timeinfo;

            time(&rawtime);
            timeinfo = localtime(&rawtime);
            printf("\nFecha De Exámen: %s", asctime(timeinfo));

            break;
        }
    }
    return (0);
}

```

### B.1.3 Sensor de pulso y oxígeno (satuometro.cpp)

Este código activa las funciones del saturómetro y lee las mediciones tomadas por el dispositivo hasta que el dato recibido por la función de pulso sea distinto de cero y el de cantidad de oxígeno en la sangre sea mayor que 10, cuando estas dos condiciones se cumplen se muestran en pantalla los valores recibidos.

```
#include "eHealth.h"
#include "stdio.h"
#include "time.h"
FILE *doc;

int cont = 0;
int x,y;
void readPulsioximeter();

void setup() {
    eHealth.initPulsioximeter();
    attachInterrupt(6, readPulsioximeter, RISING);
}

void readPulsioximeter(){

    cont ++;
    if (cont == 500) { //Get only of one 50 measures to reduce the latency
        eHealth.readPulsioximeter();
        cont = 0;
    }
}

int main (){

    doc= fopen("satu.txt","w");
    setup();
    while(1){
        x = eHealth.getOxygenSaturation();
        y = eHealth.getBPM();
        delay(100);
        if(y != 0 && x > 10){
            printf("Pulso : %d [bpm]",eHealth.getBPM());
            fprintf(doc,"%d",eHealth.getBPM());
            printf("                Saturacion(SPo2)    :    %d    %    \n\n",
eHealth.getOxygenSaturation());
            fprintf(doc,"\n%d %", eHealth.getOxygenSaturation());
            digitalWrite(2,HIGH);
            delay(100);
            time_t rawtime;
```

```

        struct tm*timeinfo;

        time(&rawtime);
        timeinfo = localtime(&rawtime);
        printf("\nFecha De Exámen: %s",asctime(timeinfo));

        break;
    }
}
return (0);
}

```

### B.1.4 Sensor de flujo respiratorio (respiracion.cpp)

Este código permite visualizar varias mediciones tomadas en un intervalo de tiempo, con ello permite generar una gráfica que representara las veces que el paciente respira y la intensidad con que lo hace.

```

#include "eHealth.h"
#include "time.h"
#include <stdio.h>
FILE *doc;

int ini;
int fin;
float t=0;
float tiempo=0;
int main (){

    doc= fopen("resp.txt","w");

    while(1){
        int air = eHealth.getAirFlow();
        //eHealth.airFlowWave(air);
        printf("%d  %.2f\n",air,tiempo);
        fprintf(doc,"%d  %.2f\n",air,tiempo);
        fin = clock();
        t= fin - ini ;
        tiempo = t/1000000;
        if(tiempo > 30.00000 ){
            break;
        }
    }
    return (0);
}

```

### B.1.5 Electrocardiograma (ecg.cpp)

Este código a diferencia del sensor de flujo respiratorio toma el valor medido y lo convierte a una variable en [mV], con este dato y el tiempo de ejecución (6 segundos), se realiza el grafico correspondiente.

```
#include "eHealth.h"
#include "time.h"
FILE *doc;
// The loop routine runs over and over again forever:
int ini;
int fin;
float t=0;
float tiempo = 0;
int main(){
    doc= fopen("ecg.txt","w");
    ini = clock();
    while(1){
        float ECG = eHealth.getECG();
        printf("%f %f\n",ECG,tiempo);
        fprintf(doc,"%f %f\n",ECG,tiempo);
        fin = clock();
        t= fin - ini ;
        tiempo = t/1000000;
        if(tiempo > 6.00000 ){
            break;
        }
    }
    return (0);
}
```

### B.1.6 Electromiograma (emg.cpp)

Este código permite recibir los valores de intensidad producidos al mover el músculo analizado, con este dato más el tiempo de ejecución se construye el gráfico de electromiograma.

```
#include "eHealth.h"
#include "time.h"
FILE *doc;

int ini;
int fin;
float t=0;
float tiempo=0;

int main(){
```

```

doc= fopen("emg.txt","w");
ini = clock();
while(1){
    int  EMG = eHealth.getEMG();
    printf("%d    %f\n",EMG,tiempo);
    fprintf(doc,"%d    %f\n",EMG,tiempo);
    fin = clock();
    t= fin - ini ;
    tiempo = t/1000000;
    if(tiempo > 20.000 ){
        break;
    }
}
return 0;
}

```

### B.1.7 Glucómetro (glucometro.cpp)

Este código accede a la memoria del glucómetro con el fin de obtener los datos almacenados y mostrar la última medición realizada.

```

#include "eHealth.h"

void setup() {

    eHealth.readGlucometer();
    delay(100);
}

int main (){
    setup();
    while(1){
        uint8_t numberOfData = eHealth.getGlucometerLength();
        delay(100);
        for (int i = eHealth.getGlucometerLength()-1; i< numberOfData; i++) {
            printf("%d \n\n",eHealth.glucoseDataVector[i].glucose);
        }
        break;
    }
}

```

### B.1.8 Esfigmomanómetro (presion.cpp)

Este código accede a la memoria del esfigmomanómetro al igual que en el glucómetro con el fin de obtener los datos almacenados y mostrar la última medición realizada.

```
#include "eHealth.h"

void setup() {
    eHealth.readBloodPressureSensor();
    delay(100);
}

void loop() {

    uint8_t numberOfData = eHealth.getBloodPressureLength();
    delay(100);

    for (int i = eHealth.getBloodPressureLength()-1; i<numberOfData; i++)
    {

        printf("%d\n", 30 + eHealth.bloodPressureDataVector[i].systolic);
        printf("%d\n", eHealth.bloodPressureDataVector[i].diastolic);
        printf("%d\n", eHealth.bloodPressureDataVector[i].pulse);
    }

    delay(10000);

}

int main (){
    setup();
    while(1){
        loop();
        break;
    }
    return (0);
}
```

### B.2 Programación interfaz gráfica

En esta sección se muestran los códigos que crean la interfaz gráfica y que permiten el inicio de sesión, la toma de datos y mediciones del paciente a evaluar y la eventual generación del reporte médico que será enviado al correo electrónico.



## B.2.1 Envío de datos via mail (a.php)

Este programa permite configurar el correo electrónico a partir del cual se envía el reporte en PDF al mail del médico encargado.

```
<?php
//Librerías para el envío de mail
include_once('phpmailer/class.phpmailer.php');
include_once('phpmailer/class.smtp.php');
//Recibir todos los parámetros del formulario
$para = $_POST['email'];
$asunto = $_POST['asunto'];
$mensaje = $_POST['mensaje'];
$sarchivo = 'e-health.pdf';
//Este bloque es importante
$mail = new PHPMailer();
$mail->IsSMTP();
$mail->SMTPAuth = true;
$mail->SMTPSecure = "ssl";
$mail->Host = "smtp.gmail.com";
$mail->Port = 465;
//Nuestra cuenta
$mail->Username = 'telemedicinapucv@gmail.com';
$mail->Password = 'TELEpucvEIE'; //Su password
//Agregar destinatario
$mail->AddAddress($para);
$mail->Subject = $asunto;
$mail->Body = $mensaje;
//Para adjuntar archivo
$mail->AddAttachment($sarchivo, $sarchivo);
$mail->MsgHTML($mensaje);
//Avisar si fue enviado o no y dirigir al index
if($mail->Send())
{
    echo'<script type="text/javascript">
        alert("Enviado Correctamente");
        window.location="http://localhost/p.html"
    </script>';
}
else{
    echo'<script type="text/javascript">
        alert("NO ENVIADO, intentar de nuevo");
        window.location="http://localhost/mail.html"
    </script>';
}
?>
```

## B.2.2 Eliminación de mediciones (borrar.php)

Este código nos permite eliminar todas las mediciones que los sensores biométricos hayan realizado, con lo cual se puede seguir analizando pacientes sin necesidad de cerrar la cuenta del usuario e iniciar nuevamente la sesión.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="es">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
  />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Plataforma E-Health</title>
    <link href="css/estilo.css" rel="stylesheet">
    <!--<link href="css/style.css" rel="stylesheet">-->
    <link href="https://fonts.googleapis.com/css?family=Ubuntu"
rel="stylesheet">
    <link rel="icon" href="src/favicon.png">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <script type="text/javascript" src="js/jquery-
1.7.1.min.js"></script>
    <link href="js/jquery-ui.css" rel="stylesheet" />
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript" src="js/jquery-ui.min.js"></script>
    <link href="css/keyboard1.css" rel="stylesheet" />
    <link href="css/estilos.css" rel="stylesheet" />
    <script type="text/javascript"
src="js/jquery.keyboard.js"></script>
    <script src="js/jquery.keyboard.extension-typing.js"
type="text/javascript"></script>
    <script type="text/javascript">
      $(document).ready(function () {
        $('#txtkeyboard1').keyboard({
          autoAccept: true
        })
        $('#txtkeyboard').keyboard({
          autoAccept: true
        })
        $('#txtkeyboard2').keyboard({
          autoAccept: true
        })
        .addTyping();
        $('#txtNumkeyboard').keyboard({
```



```

$prueb = fopen("temp.txt","w");
fclose("$prueb");
$prue = fopen("presion.txt","w");
fclose("$prue");
$prue = fopen("gluco.txt","w");
fclose("$pru");
unlink('resp.png');
copy('pdf/resp2.png','resp2.png');
unlink('ecg.png');
copy('pdf/ecg2.png','ecg2.png');
unlink('emg.png');
copy('pdf/emg2.png','emg2.png');
header('location: p.html');
}
    if(isset($_REQUEST['no'])) {

header('location: p.html');
}
?>

```

### B.2.3 Introducción de datos del paciente (datos.php)

Corresponde a la ventana donde se ingresan todos los datos del paciente, que serán tomados para luego generar el reporte médico y luego almacenar las mediciones en la base de datos del sistema.

```

<?
session_start();
include("conexion_bd.php");
?>
<html lang="es">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
/>
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <title>Plataforma E-Health</title>
        <link href="css/estilo.css" rel="stylesheet">
        <!--<link href="css/style.css" rel="stylesheet">-->
        <link href="https://fonts.googleapis.com/css?family=Ubuntu"
rel="stylesheet">
        <link rel="icon" href="src/favicon.png">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
        <script type="text/javascript" src="js/jquery-
1.7.1.min.js"></script>
        <link href="js/jquery-ui.css" rel="stylesheet" />

```

```
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/jquery-ui.min.js"></script>
<link href="css/keyboard1.css" rel="stylesheet" />
<link href="css/estilos.css" rel="stylesheet" />
<script type="text/javascript"
src="js/jquery.keyboard.js"></script>
<script src="js/jquery.keyboard.extension-typing.js"
type="text/javascript"></script>
<script type="text/javascript">
$(document).ready(function () {
    $('#txtkeyboard1').keyboard({
        autoAccept: true
    })
    $('#txtkeyboard').keyboard({
        autoAccept: true
    })
    $('#txtkeyboard2').keyboard({
        autoAccept: true
    })
    $('#txtkeyboard3').keyboard({
        autoAccept: true
    })
    $('#txtkeyboard4').keyboard({
        autoAccept: true
    })
    $('#txtkeyboard5').keyboard({
        autoAccept: true
    })
    $('#txtkeyboard6').keyboard({
        autoAccept: true
    })
        .addTyping();
    $('#txtNumkeyboard').keyboard({
        layout: 'num',
        restrictInput: true,
        preventPaste: true,
        autoAccept: true
    })
        .addTyping();
});
</script>
<body id="body_login">
<?php
$hora = date("H:i:s");
$fecha = date("j/n/Y");
$txt = utf8_decode($_GET['name']);
$txt1 = $_GET['rut'];
```

```

$txt2 = $_GET['edad'];
$txt3 = utf8_decode($_GET['servicio']);
$txt4 = utf8_decode($_GET['procedencia']);
$txt5 = utf8_decode($_GET['medico']);
$ar = fopen("datos1.txt", "w+");
$ar2 = fopen("datos2.txt", "w+");
fwrite($ar2, $fecha);
fwrite($ar2, "\n");
fwrite($ar2, "\n");
fwrite($ar2, $hora);
fwrite($ar2, "\n");
fwrite($ar2, "\n");
fwrite($ar, 'Nombre de paciente: ');
fwrite($ar, $txt);
fwrite($ar, "\n");
fwrite($ar, "\n");
fwrite($ar, 'RUN: ');
fwrite($ar, $txt1);
fwrite($ar, "\n");
fwrite($ar, "\n");
fwrite($ar, 'Edad: ');
fwrite($ar, $txt2);
fwrite($ar, "\n");
fwrite($ar, "\n");
fwrite($ar, 'Servicio: ');
fwrite($ar, $txt3);
fwrite($ar, "\n");
fwrite($ar, "\n");
fwrite($ar, 'Procedencia: ');
fwrite($ar, $txt4);
fwrite($ar, "\n");
fwrite($ar, "\n");
fwrite($ar, 'Medico a cargo: ');
fwrite($ar, $txt5);
fclose($ar);
fclose($ar1);
if (isset($_REQUEST['aplicar'])) {
    $nombre = utf8_encode($txt);
    $servicio = utf8_encode($txt3);
    $procedencia = utf8_encode($txt4);
    $medico = utf8_encode($txt5);
    mysql_query("insert                into                pacientes
(nombre,rut,edad,servicio,procedencia,medico)                values
('$nombre','$txt1','$txt2','$txt3','$txt4','$txt5')");
    $_SESSION["id_paciente"] = mysql_insert_id();
    header('location: p.html');
}

```

```

?>

<div id="login2">
  <h1>Datos del Paciente</h1>
  <form method="get" action="datos.php">
    <div id="datos">
      <div id="dato">
        Nombre:<br />
        <input type="text" name="name" id="txtkeyboard"
required><br />
      </div>
      <div id="dato">
        RUT:<br />
        <input type="text" name="rut" id="txtkeyboard1" required
maxlength="10"><br />
      </div>
      <div id="dato">
        Edad:<br />
        <input type="text" name="edad" id="txtkeyboard2" required
maxlength="2"><br />
      </div>
    </div>
    <div id="datos">
      <div id="dato">
        Servicio:<br />
        <input type="text" name="servicio" id="txtkeyboard3"
required><br />
      </div>
      <div id="dato">
        Procedencia:<br />
        <input type="text" name="procedencia"
id="txtkeyboard4"/><br />
      </div>
      <div id="dato">
        Médico:<br />
        <input type="text" name="medico" id="txtkeyboard5"
value="<? echo $_SESSION["dr"]; ?>" required><br />
      </div>
    </div>
    <br />
    <input type="submit" value="Ingresar Datos" name="aplicar">
  </form>
  <br />
  <form method="post" action="resultados.php">
    Búsqueda de Paciente por RUT:
    <input type="text" name="rut" required id="txtkeyboard6">
    <input type="submit" value="Búsqueda" name="busqueda">
  </form>

```

```

    </form>
  </div>
</body>
</html>

```

### B.2.4 Guardado de las mediciones (datos2.php)

Con este código se guardan los valores obtenidos en las mediciones tomadas por los sensores biométricos, para luego traspasarlos al reporte en PDF que se enviara al correo electrónico.

```

<html lang="es">
  <head>
    <title>Plataforma E-Health</title>
    <link href="css/estilo.css" rel="stylesheet">
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
  />
    <meta http-equiv="Refresh" content="1">
  </head>
  <body id="aw">
    <form method="post" action="sat.php" align="left" id="data">
      Pulso:<br />
      <input type="text" style="text-align: center" name="saturacion"
size="15" value="<?php
      $fp = fopen("satu.txt", "r");
      if (isset($_REQUEST['procesar'])) {
        exec("./satuometro", $output);
        foreach ($output as $out) {
#echo $out."<br/>";
        }
      }
      $fichero = "satu.txt";
      $filas = file($fichero);
      $ultima_linea5 = $filas[count($filas) - 1];
      $ultima_linea6 = $filas[count($filas) - 2];
      $ultima_linea7 = $filas[count($filas) - 3];
      echo "$ultima_linea6" . "\n";
      ?>">[bpm]<br/>
      Oxigeno:<br/>
      <input type="text" style="text-align: center" name="saturacion"
size="15" value="<?php
      $fichero = "satu.txt";
      $filas = file($fichero);
      $ultima_linea5 = $filas[count($filas) - 1];
      $ultima_linea6 = $filas[count($filas) - 2];
      $ultima_linea7 = $filas[count($filas) - 3];
      echo "$ultima_linea5" . "\n";
      ?>">[SPo2]<br/>

```



```

    Temperatura:<br/>
    <input type="text" style="text-align: center" name="saturacion"
size="15" value="<?php
    $fichero = "temp.txt";
    $filas = file($fichero);
    $ultima_linea5 = $filas[count($filas) - 1];
    $ultima_linea6 = $filas[count($filas) - 2];
    $ultima_linea7 = $filas[count($filas) - 3];
    echo "$ultima_linea5" . "\n";
    ?>">[°C]
    <br>
    Glucosa:<br/>
    <input type="text" style="text-align: center" name="saturacion"
size="15" value="<?php
    $fichero = "gluco.txt";
    $filas = file($fichero);
    $ultima_linea5 = $filas[count($filas) - 1];
    $ultima_linea6 = $filas[count($filas) - 2];
    $ultima_linea7 = $filas[count($filas) - 3];
    echo "$ultima_linea6" . "\n";
    ?>">[mg/dL]<br>
    P.Diastolica:<br/>
    <input type="text" style="text-align: center" name="saturacion"
size="15" value="<?php
    $fichero = "presion.txt";
    $filas = file($fichero);
    $ultima_linea = $filas[count($filas) - 1];
    $ultima_linea2 = $filas[count($filas) - 2];
    $ultima_linea3 = $filas[count($filas) - 3];
    $ultima_linea4 = $filas[count($filas) - 4];
    echo "$ultima_linea2" . "\n";
    ?>">[mmHg]<br>
    P.Sistolica:<br/>
    <input type="text" style="text-align: center" name="saturacion"
size="15" value="<?php
    $fichero = "presion.txt";
    $filas = file($fichero);
    $ultima_linea = $filas[count($filas) - 1];
    $ultima_linea2 = $filas[count($filas) - 2];
    $ultima_linea3 = $filas[count($filas) - 3];
    $ultima_linea4 = $filas[count($filas) - 4];

    echo "$ultima_linea3" . "\n";
    ?>">[mmHg]<br>
    <div style="display: none;">
    Pulso:<br/>

```

```

        <input type="text" style="text-align: center" name="saturacion"
size="15" value="<?php
        $fichero = "presion.txt";
        $filas = file($fichero);
        $ultima_linea = $filas[count($filas) - 1];
        $ultima_linea2 = $filas[count($filas) - 2];
        $ultima_linea3 = $filas[count($filas) - 3];
        $ultima_linea4 = $filas[count($filas) - 4];

        echo "$ultima_linea", "\n";
        ?>">[bpm]<br>
    </div>
</form>
</body>
</html>

```

### B.2.5 Ventana gráficas (graficas.php)

Código que nos muestra la ventana con los botones de los tres exámenes gráficos que se realizan, la cual después nos lleva a la visualización de cada uno de ellos y nos permite volver al menú principal.

```

<html lang="es">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
  />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Plataforma E-Health</title>
    <link href="css/estilo.css" rel="stylesheet">
    <!--<link href="css/style.css" rel="stylesheet">-->
    <link href="https://fonts.googleapis.com/css?family=Ubuntu"
rel="stylesheet">
    <link rel="icon" href="src/favicon.png">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <script type="text/javascript" src="js/jquery-
1.7.1.min.js"></script>
    <link href="js/jquery-ui.css" rel="stylesheet" />
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript" src="js/jquery-ui.min.js"></script>
    <link href="css/keyboard1.css" rel="stylesheet" />
    <link href="css/estilos.css" rel="stylesheet" />
    <script type="text/javascript"
src="js/jquery.keyboard.js"></script>
    <script src="js/jquery.keyboard.extension-typing.js"
type="text/javascript"></script>

```

```

<body id="body_login">
  
  <H2 align="center"> Graficas </H2>
  <br/><br/>
<style>
  /* options with image */
  .boton {
    width: 200px;
  }
  .opciones {
    overflow:hidden;
    text-align: center;
    margin:0;
    background:;
  }
  .opcion {
    display:inline-table;
    border:0px solid #ccc;
    padding:2px;
    height:10px;
    width:250px;
    margin:3px;
  }
</style>
<div class="opciones">
  <div class="opcion">
    <form method="post" action="graficas.php"><input type="submit"
name="electro" value = "Electrocardiograma" class="boton">
    <br/>    <?php
    if (isset($_REQUEST['electro'])) {
      exec("./ecg", $output);
      exec("php5 crearecg.php", $output);
      header('location: graficaecg.php');
    }
    ?>
  </form>
</div>
  <div class="opcion">
    <form method="post" action="graficas.php"><input type="submit"
name="electromiograma" value = "Electromiograma" class="boton">
    <br />
    <?php
    if (isset($_REQUEST['electromiograma'])) {

      exec("./EMG", $output);
      exec("php5 crearemg.php", $output);
      header('location: graficaemg.php');

```

```

    }
    ?>
  </form> </div><br /><br />
  <div class="opcion">
    <form method="post" action="graficas.php"><input type="submit"
name="respiracion" value = "Respiracion" class="boton">
    <br />
    <?php
    if (isset($_REQUEST['respiracion'])) {
      exec("./respiracion", $output);
      exec("php5 crearresp.php", $output1);
      header('location: graficaresp.php');
    }
    ?>
  </form> </div><br /><br/><br /><br/><br /><br/>
  <div class="opcion">
    <form method="post" action="p.html"><input type="submit"
name="Menu Principal" value = "Menu Principal" class="boton">
    <br />
  </form> </div>
</div>
</body>
</html>

```

### B.2.6 Ventana de observaciones (observacion.php)

Se despliega cuando queremos hacer una observación importante sobre el estado del paciente según el resultado de los exámenes realizados, ésta se adjunta en el reporte en PDF.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="es">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
  />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Plataforma E-Health</title>
    <link href="css/estilo.css" rel="stylesheet">
    <!--<link href="css/style.css" rel="stylesheet"-->
    <link href="https://fonts.googleapis.com/css?family=Ubuntu"
rel="stylesheet">
    <link rel="icon" href="src/favicon.png">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <script type="text/javascript" src="js/jquery-
1.7.1.min.js"></script>

```

```

<link href="js/jquery-ui.css" rel="stylesheet" />
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/jquery-ui.min.js"></script>
<link href="css/keyboard1.css" rel="stylesheet" />
<link href="css/estilos.css" rel="stylesheet" />
<script
                                                                    type="text/javascript"
src="js/jquery.keyboard.js"></script>
<script
                                                                    src="js/jquery.keyboard.extension-typing.js"
type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function () {
        $('#txtkeyboard1').keyboard({
            autoAccept: true
        })
        $('#txtkeyboard').keyboard({
            autoAccept: true
        })
        $('#txtkeyboard2').keyboard({
            autoAccept: true
        })
        .addTyping();
        $('#txtNumkeyboard').keyboard({
            layout: 'num',
            restrictInput: true,
            preventPaste: true,
            autoAccept: true
        })
        .addTyping();
    });
</script>
<body id="body_login">
    
    <H2> Observaciones: </H2>
<style type="text/css">
    .boton{
        font-size:17px;
        font-family:Verdana,Helvetica;
        font-weight:bold;
        color:white;
        background:#638cb5;
        border:10px;
        width:200px;
        height:30px;
    }
    .boton1{
        font-size:17px;
        font-family:Verdana,Helvetica;

```

```

        font-weight:bold;
        color:white;
        background:#638cb5;
        border:0px;
        width:25px;
        height:25px;
    }
</style>
<style>
    /* options with image */
    .opciones {
        overflow:hidden;
        text-align: center;
        margin:auto;
        background;;
    }
    .opcion {
        display:inline-table;
        border:0px solid #ccc;
        padding:15px;
        height:20px;
        width:10px;
        margin:3px;
    }
</style>
<div class="opciones">
    <div class="opcion">
        <form action="observacion.php" method="get" align="center">
            <TEXTAREA rows="10" wrap="hard" cols="50" name="texto"
id="txtkeyboard"></TEXTAREA>
<BR /><BR />
<input type="submit" name="aplicar" value="Agregar Observación"/>
</form>
<br /><br />
    <form method="post" action="p.html">
        <input type="submit" name="Menu Principal" value = "Menu
Principal">
    </form>
</div>
</div>
<?php
$hora = date("H:i:s");
$fecha = date("j/n/Y");
$txt = $_GET['texto'];
$ar = fopen("datos14.txt", "w+");
fwrite($ar, $txt);
fwrite($ar, "\n");

```

```

    fwrite($ar, "\n");
    fclose($ar);
    if (isset($_REQUEST['aplicar'])) {
        header('location: p.html');
    }
    ?>
</body>
</html>

```

### B.2.7 Configuración y creación de PDF (pdf.php)

La creación del reporte médico en PDF, se especifica mediante el código que se adjunta a continuación, con él se modifica la plantilla y se traspasan los datos que fueron añadidos con anterioridad en la interfaz gráfica.

```

<?
session_start();
?>
<html>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
/>
    <?php
        include("conexion_bd.php");
        /*$f = fopen("presion.txt", 'r');
        //$presion=fread($f,filesize($file));
        $presion = file_get_contents('presion.txt');
        fclose($f);*/
        $lineas = array();
        $fp = fopen('presion.txt', 'r');
        while (!feof($fp)) {
            $linea = fgets($fp);
            $linea = trim($linea);
            $lineas[] = $linea;
        }
        fclose($fp);
        $diastolica = $lineas[0];
        $sistolica = $lineas[1];
        $f = fopen("temp.txt", 'r');
        //$temperatura=fread($f,filesize($file));
        $temperatura = file_get_contents('temp.txt');
        fclose($f);
        $f = fopen("gluco.txt", 'r');
        //$glucosa=fread($f,filesize($file));
        $glucosa = file_get_contents('gluco.txt');
        fclose($f);
        /* $f=fopen("satu.txt",'r');
        //fila 1 pulso

```

```

        //fila 2 saturacion
        $saturacion=fread($f, filesize($file));
        fclose($f); */
    $lineas = array();
    $fp = fopen('satu.txt', 'r');
    while (!feof($fp)) {
        $linea = fgets($fp);
        $linea = trim($linea);
        $lineas[] = $linea;
    }
    fclose($fp);
    $pulso = $lineas[0];
    $saturacion = $lineas[1];
    $f = fopen("datos14.txt", 'r');
    //$observaciones=fread($f, filesize($file));
    $observaciones = file_get_contents('datos14.txt');
    fclose($f);
    $id_pacientes = $_SESSION["id_paciente"];
    $id_doctores = $_SESSION["id_dr"];

    $fecha=date("Y-m-d");
    $observaciones2=utf8_encode($observaciones);
    mysql_query("insert into examenes (id_pacientes, id_doctores,
diastolica, sistolica, temperatura, glucosa, saturacion, pulso,
observaciones, fecha) values (\"
.
\"'$id_pacientes', '$id_doctores', '$diastolica', '$sistolica', '$temperatu
ra', '$glucosa', '$saturacion', '$pulso', '$observaciones2', '$fecha')");
    ob_end_clean();
    require('fpdf.php');
    class PDF extends FPDF {
//Cabecera de página
        function Header() {
//Logo
//Arial bold 15
            $this->SetFont('Arial', 'B', 15);
//Movernos a la derecha
            $this->Cell(80);
//Título
            $this->Cell(60, 10, 'Plataforma E-Health', 1, 0, 'C');
//Salto de línea
            $this->Ln(1);
        }
//Pie de página
        function Footer() {
            $f2 = fopen('firma.txt', 'r');
            $txt9 = fread($f2, filesize('firma.txt'));

```



```

        fclose($f2);
//Posición: a 1,5 cm del final
        $this->SetY(-20);
//Arial italic 8
        $this->SetFont('Arial', 'I', 15);
//Número de página
        $this->MultiCell(0, 2, $txt9, 0, 0);
        $this->SetY(-15);
        $this->SetFont('Arial', 'I', 8);
        $this->Cell(0, 10, 'Page ' . $this->PageNo(), 0, 0, 'C');
    }
function TituloArchivo($num, $label) { {
    $f1 = fopen('datos1.txt', 'r');
    $txt1 = fread($f1, filesize('datos1.txt'));
    fclose($f1);
    $this->SetY(30);
    //Arial 12
    $this->SetFont('Arial', '', 12);
//Color de fondo
    $this->SetFillColor(200, 220, 255);
//Título
    $this->MultiCell(0, 2, $txt1, 'C');
    $this->Ln(2);
    $this->Cell(0, 6, "Examen $num: $label", 0, 1, 'L', true);
//Salto de línea
    } {
    $f2 = fopen('datos2.txt', 'r');
    $txt2 = fread($f2, filesize('datos2.txt'));
    fclose($f2);
    $this->SetY(30);
    $this->Setx(150);
    $this->MultiCell(0, 2, $txt2, 'C');
    }
    $this->Ln(25);
}
function CuerpoArchivo($file) {
//Leemos el fichero
    $f = fopen($file, 'r');
    /*
        $pdf->ImprimirArchivo(1, 'Presion', 'presion.txt');
        $pdf->prueba(2, 'Temperatura ', 'temp.txt');
        $pdf->ImprimirArchivo(3, 'Glucosa ', 'gluco.txt');
        $pdf->ImprimirArchivo(4, 'Saturacion ', 'satu.txt');
        $pdf->ImprimirArchivo(0, 'Observaciones
del
paciente', 'datos14.txt');
    */
    if ($file == "presion.txt") {

```

```
    $txt = "Presion Diastolica: ";
}
elseif ($file == "temp.txt") {
    $txt = "Temperatura: ";
}
elseif ($file == "gluco.txt") {
    $txt = "Glucosa: ";
}
elseif ($file == "satu.txt") {
    $txt = "Pulso: ";
}
elseif ($file == "datos14.txt") {
    $txt = "Observaciones: ";
}
//LECTURA
if ($file == "presion.txt") {
    $lineas = array();
    while (!feof($f)) {
        $linea = fgets($f);
        $linea = trim($linea);
        $lineas[] = $linea;
    }
    $txt.=$lineas[0]." [mmHg]\n";
    $txt.="Presion Sistolica: ";
    $txt.=$lineas[1]." [mmHg]";
}
elseif ($file == "satu.txt") {
    $lineas = array();
    while (!feof($f)) {
        $linea = fgets($f);
        $linea = trim($linea);
        $lineas[] = $linea;
    }
    $txt.=$lineas[0]." [bpm]\n";
    $txt.="Saturacion: ";
    $txt.=$lineas[1]." [SPo2]";
}
else {
    //lectura para cualquiera
    $txt.=fread($f, filesize($file));
}
if ($file == "temp.txt") {
    $txt.=" [°C]";
}
elseif ($file == "gluco.txt") {
    $txt.=" [mg/dL]";
}
}
```

```

        fclose($f);
//Times 12
        $this->SetFont('Times', '', 18);
//Imprimimos el texto justificado
        $this->MultiCell(0, 5, $txt);
//Salto de línea
        $this->Ln();
    }
    function ImprimirArchivo($num, $title, $file) {
        $this->AddPage();
        $this->TituloArchivo($num, $title);
        $this->CuerpoArchivo($file);
    }
    function prueba($num, $title, $file) {
        $this->AddPage();
        $this->TituloArchivo($num, $title);
        $this->CuerpoArchivo($file);
    }
    function prueba1($num, $title, $file) {
        $this->AddPage('L');
        $this->TituloArchivo($num, $title);
        $this->CuerpoArchivo($file);
        $this->Image('resp2.png', 22, 65, 250);
    }
    function prueba2($num, $title, $file) {
        $this->AddPage('L');
        $this->TituloArchivo($num, $title);
        $this->CuerpoArchivo($file);
        $this->Image('ecg2.png', 22, 65, 250);
    }
    function prueba3($num, $title, $file) {
        $this->AddPage('L');
        $this->TituloArchivo($num, $title);
        $this->CuerpoArchivo($file);
        $this->Image('emg2.png', 22, 65, 250);
    }
}
$pdf = new PDF();
$title = 'Reporte Medico';
$pdf->SetTitle($title);
$pdf->SetY(65);
$pdf->SetFont('Times', '', 25);
$pdf->ImprimirArchivo(1, 'Presion ', 'presion.txt');
$pdf->prueba(2, 'Temperatura ', 'temp.txt');
$pdf->ImprimirArchivo(3, 'Glucosa ', 'gluco.txt');
$pdf->ImprimirArchivo(4, 'Pulso y Saturacion ', 'satu.txt');
$pdf->prueba1(5, 'Respiracion ');

```

```

$pdf->prueba2(6, 'Electrocardiograma ');
$pdf->prueba3(7, 'Electromiograma ');
$pdf->ImprimirArchivo(8, 'Observaciones del paciente ',
'datos14.txt');

$pdf->Output('e-health.pdf', 'F');
header('location: maill.html');
?>
</html>

```

## B.2.8 Ventana principal de inicio de sesión (pincipal.php)

Esta es nuestra primera interacción con el dispositivo portátil, donde debemos poner nuestro nombre de usuario y contraseña correctas para que el sistema nos deje ingresar al menú principal y así poder realizar las mediciones.

```

<?php
include("conexion_bd.php");
file_put_contents("satu.txt", "");
file_put_contents("temp.txt", "");
file_put_contents("presion.txt", "");
file_put_contents("gluco.txt", "");
unlink('resp.png');
copy('pdf/resp2.png', 'resp2.png');
unlink('ecg.png');
copy('pdf/ecg2.png', 'ecg2.png');
unlink('emg.png');
copy('pdf/emg2.png', 'emg2.png');
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="es">
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Plataforma E-Health</title>
    <link href="css/estilo.css" rel="stylesheet">
    <!--<link href="css/style.css" rel="stylesheet">-->
    <link href="https://fonts.googleapis.com/css?family=Ubuntu"
rel="stylesheet">
    <link rel="icon" href="src/favicon.png">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <script type="text/javascript" src="js/jquery-
1.7.1.min.js"></script>

```

```

    <link href="js/jquery-ui.css" rel="stylesheet" />
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script          type="text/javascript"          src="js/jquery-
ui.min.js"></script>
    <link href="css/keyboard1.css" rel="stylesheet" />
    <link href="css/estilos.css" rel="stylesheet" />
    <script
src="js/jquery.keyboard.js"></script>
    <script          src="js/jquery.keyboard.extension-typing.js"
type="text/javascript"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $('#txtkeyboard1').keyboard({
                autoAccept: true
            })
            $('#txtkeyboard').keyboard({
                autoAccept: true
            })
            $('#txtkeyboard2').keyboard({
                autoAccept: true
            })
                .addTyping();
            $('#txtNumkeyboard').keyboard({
                layout: 'num',
                restrictInput: true,
                preventPaste: true,
                autoAccept: true
            })
                .addTyping();
        });
    </script>
<body id="body_login">
    
    <div id="login">
        <?php
        $name = $_POST["name"];
        $pass = $_POST["pass"];
        if (!isset($name) && !isset($pass)) {
            ?>
        <h1>Ingreso al Sistema</h1>
        <form method="post">
            Usuario:<br />
            <input type="text" name="name" id="txtkeyboard1"
required><br />
            Contraseña:<br />
            <input type="password" name="pass" id="txtkeyboard"
required><br />

```

```

        <br />
        <input type="submit" value="Entrar">
    </form>
    <?php
    }
    else {
        $accion = mysql_query("select * from doctores where
usuario='$name' and pass='$pass'");
        if (mysql_num_rows($accion) != 0) { // si hay
coincidencias
            session_start();
            while ($fila= mysql_fetch_array($accion)) {
                $_SESSION["dr"]=$fila["nombre"];
                $_SESSION["id_dr"]=$fila["id_doctores"];
            }
            header('location: datos.php');
        }
        else {
            echo " <h1>Acceso no autorizado.</h1><a
href='javascript:window.history.go(-1) '>Volver</a></html>";
        }
    }
    ?>
</div>
</body>
</html>

```

### B.2.9 Creación del reporte (reporte.php)

Este código nos despliega la ventana del reporte médico, permite crear el PDF que luego se enviara vía mail. En esta sección aparece la dirección del correo electrónico a la que queremos enviar el mail, el asunto y el mensaje con la acotación que queremos realizar.

```

<html>
<head>
    <title>Reporte </title>
    <link
href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/ui-
lightness/jquery-ui.css" rel="stylesheet" />
<script
type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.6/jquery.min.js"></
script>
<script
type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-
ui.min.js"></script>
<link href="css/keyboard1.css" rel="stylesheet" />
<script type="text/javascript" src="js/jquery.keyboard.js"></script>

```

```

        <script src="js/jquery.keyboard.extension-typing.js"
type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function () {
        $('#txtkeyboard1').keyboard({
            autoAccept:true

        })
        $('#txtkeyboard').keyboard({
            autoAccept:true

        })
        .addTyping();
        $('#txtNumkeyboard').keyboard({
            layout: 'num',
            restrictInput: true,
            preventPaste: true,
            autoAccept: true
        })
        .addTyping();
    });
</script>
</head>
<body>
<table>
<tr>
<td><b>        </b></td><td>
            <form method="post" action="pdf.php">
                <input type="submit" value="Reporte" class = "boton">

                    <input type="text" name="caja" id="txtkeyboard1"
/></form></td>
</table>
<td><b>        </b></td><td>
            <form method="post" action="pdf.php">
                <input type="submit" value="Reporte" class = "boton">

                    <input type="text" name="caja" id="txtkeyboard" /></form></td>
</table>
</body>
</html>

```

## B.2.10 Ventana de menú principal (p.html)

Despliega la ventana de menú principal, donde se encuentran todos los botones y aparecen en pantalla todas las mediciones correspondientes. Cuenta con las subdivisiones que nos redireccionan a todas las demás ventanas de la interfaz.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="es">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Plataforma E-Health</title>
    <link href="css/estilo.css" rel="stylesheet">
  </head>
  <frameset rows="10%,*" cols="25%,30%,20%" SCROLLING="no"
  frameborder="0" border="0" framespacing="0" >
    <frame src="iz.php"frameborder="0" marginwidth="0" SCROLLING="no"
    marginheight="0" BORDER="0">
    <frame src="titulo.php"frameborder="0" marginwidth="0" SCROLLING="no"
    marginheight="0" BORDER="0">
    <frame src="de.php"frameborder="0" marginwidth="0" SCROLLING="no"
    marginheight="0" BORDER="0">
    <frameset cols="80%" rows="30%,30%,30%,30%,80%" SCROLLING="no"
    frameborder="0" border="0" framespacing="0" >
      <frame src="sat1.php"frameborder="0"SCROLLING="no">
      <frame src="glu1.php"frameborder="0"SCROLLING="no">
      <frame src="pre1.php"frameborder="0" name="fill_me"SCROLLING="no">
      <frame scrolling="no" frameborder="0" src="temp0.php">
      <frame src="gra.php" frameborder="0" SCROLLING="no">
    </frameset>
    <frameset cols="80%" rows="70%" SCROLLING="no" frameborder="0"
    border="0" framespacing="0" >
      <frame src="datos2.php"frameborder="0" SCROLLING="no">
    </frameset>
    <frameset cols="80%" rows="78%,20%" SCROLLING="no" frameborder="0"
    border="0" framespacing="0" >
      <frame src="otros.php">
      <frame src="hora.php" >
    </frameset>
  </html>
```

## 6.2.2 Datos almacenados en el sistema (resultados.php)

Este programa permite visualizar dos ventanas, primero la que almacena todos los exámenes realizados al mismo paciente y que se busca por el RUT y la segunda es la observación de los exámenes que se solicita ver dependiendo del día en que fue realizado.



```

<?
session_start();
include("conexion_bd.php");
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="es">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
  />
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Plataforma E-Health</title>
    <link href="css/estilo.css" rel="stylesheet">
    <!--<link href="css/style.css" rel="stylesheet"-->
    <link href="https://fonts.googleapis.com/css?family=Ubuntu"
rel="stylesheet">
    <link rel="icon" href="src/favicon.png">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <script type="text/javascript" src="js/jquery-
1.7.1.min.js"></script>
    <link href="js/jquery-ui.css" rel="stylesheet" />
    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript" src="js/jquery-ui.min.js"></script>
    <link href="css/keyboard1.css" rel="stylesheet" />
    <link href="css/estilos.css" rel="stylesheet" />
    <script type="text/javascript"
src="js/jquery.keyboard.js"></script>
    <script src="js/jquery.keyboard.extension-typing.js"
type="text/javascript"></script>
  <body id="body_login">
    
    <div id="login2">
      <h1>Resultados de Búsqueda</h1>
      <?
      $rut = $_POST["rut"];
      if (!$rut) {
        $rut = $_GET["rut"];
      }
      $id_examenes = $_GET["id"];
      $pagina = $_GET["pagina"];
      if ($rut) {
        $accion = mysql_query("select * from pacientes where
rut='$rut'");
        if (mysql_num_rows($accion) == 0) {

```

```

        echo "Ningún resultado encontrado";
    }
    else {
        $numero = 0;
        while ($fila = mysql_fetch_array($accion)) {
            $id_pacientes = $fila["id_pacientes"];
            $accion2 = mysql_query("select * from examenes where
id_pacientes='$id_pacientes'");
            $numero += mysql_num_rows($accion2);
        }
        $accion = mysql_query("select * from pacientes where
rut='$rut'");
        $numero_paginas = $numero / 10;
        $numero_paginas = explode(".", $numero_paginas);
        $paginas = $numero_paginas[0];
        $decimales = $numero_paginas[1];
        if ($decimales > 0) {
            $paginas++;
        }
        $contador = 0;
        if (!$pagina) {
            $pagina = 1;
        }
        $pag_prev = $pagina - 1;
        $pag_sig = $pagina + 1;
        $conteo_final = $pagina * 10;
        $conteo_inicial = $conteo_final - 10;
        echo "<br />";
        //Previa
        if ($pagina != 1) {
            echo
href='resultados.php?rut=$rut&pagina=$pag_prev'>&lt;</a> ";
        }
        else {
            echo "&lt; ";
        }
        //Paginas
        if ($paginas > 1) {
            for ($i = 1; $i <= $paginas; $i++) {
                if ($i != $pagina) {
                    echo
href='resultados.php?rut=$rut&pagina=$i'>$i</a> ";
                }
                else {
                    echo "$i";
                }
                if ($i < $paginas) {

```



```

        $accion3 = mysql_query("select * from pacientes where
id_pacientes='$id_pacientes'");
        while ($fila3 = mysql_fetch_array($accion3)) {
            $nombre_paciente = utf8_decode($fila3["nombre"]);
        }
        $id_doctores = $fila2["id_doctores"];
        $accion3 = mysql_query("select * from doctores where
id_doctores='$id_doctores'");
        while ($fila3 = mysql_fetch_array($accion3)) {
            $nombre_doctor = utf8_decode($fila3["nombre"]);
        }
        $diastolica = $fila2["diastolica"];
        $sistolica = $fila2["sistolica"];
        $temperatura = $fila2["temperatura"];
        $glucosa = $fila2["glucosa"];
        $saturacion = $fila2["saturacion"];
        $pulso = $fila2["pulso"];
        $observaciones = utf8_decode($fila2["observaciones"]);
    }
    ?>
<div id="datos">
    <div id="dato" style="text-align: left;">
        Nombre Paciente:<br />
        Nombre Doctor:<br />
        Presión Diastólica: <br />
        Presión Sistólica: <br />
        Temperatura: <br />
        Glucosa: <br />
        Saturación: <br />
        Pulso: <br />
    </div>
    <div id="dato" style="text-align: left;">
        <? echo $nombre_paciente; ?><br />
        <? echo $nombre_doctor; ?><br />
        <? echo $diastolica; ?>[mmHg]<br />
        <? echo $sistolica; ?>[mmHg] <br />
        <? echo $temperatura; ?>[°C] <br />
        <? echo $glucosa; ?>[mg/dL] <br />
        <? echo $saturacion; ?>[SPo2] <br />
        <? echo $pulso; ?>[mmHg] <br />
    </div>
</div>
<div id="datos" style="text-align: left;">
    Observaciones: <? echo $observaciones; ?>
</div>
<?
echo "<br /><br /><a href='datos.php'>Volver</a>";

```

```

    }
    ?>
  </div>
</body>
</html>

```

### B.3 Programación y conexión a la base de datos

A continuación, se muestra como se creó la base de datos en phpMyAdmin, la cual se hace mediante tablas permitiendo exportar todas las configuraciones a un código en SQL con el fin de utilizar todo el trabajo realizado en otro servidor. Además, se muestra como se conecta la base de datos con la interfaz gráfica desarrollada en PHP.

#### B.3.1 Base de datos (ehealth\_bd.sql)

A continuación, se muestra cual es el código SQL que permite exportar phpMyAdmin desde su aplicación en el servidor y cuenta con todas las características que se les asignan a los datos a almacenar y además guarda los cambios realizados en la base de datos.

```

-- phpMyAdmin SQL Dump
-- version 3.4.11.1deb2+deb7u8
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Dec 06, 2017 at 06:25 PM
-- Server version: 5.5.58
-- PHP Version: 5.4.45-0+deb7u11
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
--
-- Database: `ehealth_bd`
--
-- -----
--
-- Table structure for table `doctores`
--
CREATE TABLE IF NOT EXISTS `doctores` (
  `id_doctores` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(200) NOT NULL,
  `usuario` varchar(20) NOT NULL,
  `pass` varchar(200) NOT NULL,
  PRIMARY KEY (`id_doctores`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

```

```

--
-- Dumping data for table `doctores`
--
INSERT INTO `doctores` (`id_doctores`, `nombre`, `usuario`, `pass`)
VALUES
(1, 'Doctor', 'user1', 'pass');
-----
--
-- Table structure for table `examenes`
--
CREATE TABLE IF NOT EXISTS `examenes` (
  `id_examenes` int(11) NOT NULL AUTO_INCREMENT,
  `id_pacientes` int(11) NOT NULL,
  `id_doctores` int(11) NOT NULL,
  `diastolica` varchar(10) NOT NULL,
  `sistolica` varchar(10) NOT NULL,
  `temperatura` varchar(10) NOT NULL,
  `glucosa` varchar(10) NOT NULL,
  `saturacion` varchar(10) NOT NULL,
  `pulso` varchar(10) NOT NULL,
  `observaciones` text NOT NULL,
  `fecha` date NOT NULL,
  PRIMARY KEY (`id_examenes`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=25 ;
--
-- Dumping data for table `examenes`
--
INSERT INTO `examenes` (`id_examenes`, `id_pacientes`, `id_doctores`,
`diastolica`, `sistolica`, `temperatura`, `glucosa`, `saturacion`,
`pulso`, `observaciones`, `fecha`) VALUES
(1, 1, 1, '111', '74', '35.81', '79 \r\n\r\n', '98', '69', 'Normal',
'2017-11-30'),
(2, 2, 1, '115', '71', '36.99', '80 \r\n\r\n', '97', '90', 'Normal',
'2017-12-06'),
(3, 3, 1, '128', '81', '35.79', '90 \r\n\r\n', '97', '76', 'Normal\r\n',
'2017-11-30'),
(4, 4, 1, '135', '80', '36.37', '104\r\n\r\n', '96', '72', 'Estado
normal, a considerar presion un poco alta.\r\n', '2017-12-06'),
(5, 5, 1, '107', '74', '36.83', '103 \r\n\r\n', '98', '70', 'Estado
normal, todos los signos vitales en orden.\r\n', '2017-12-06'),
(6, 6, 1, '', '', '', '94', '86', '\r\n', '2017-12-06');
-----
--
-- Table structure for table `pacientes`
--
CREATE TABLE IF NOT EXISTS `pacientes` (
  `id_pacientes` int(11) NOT NULL AUTO_INCREMENT,

```

```
`nombre` varchar(200) NOT NULL,  
`rut` varchar(10) NOT NULL,  
`edad` int(2) NOT NULL,  
`servicio` varchar(200) NOT NULL,  
`procedencia` varchar(200) NOT NULL,  
`medico` varchar(200) NOT NULL,  
PRIMARY KEY (`id_pacientes`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=30 ;  
--  
-- Dumping data for table `pacientes`  
--  
INSERT INTO `pacientes` (`id_pacientes`, `nombre`, `rut`, `edad`,  
`servicio`, `procedencia`, `medico`) VALUES  
(1, 'Esteban Perez', '18561579-0', 24, 'CONSULTORIO', 'VALPARAISO',  
'Doctor'),  
(2, 'Esteban Perez', '18561579-0', 24, 'CONSULTORIO', 'SALAMANCA',  
'Doctor'),  
(3, 'Esteban Perez', '18561579-0', 24, 'CONSULTORIO', 'SALAMANCA',  
'Gloria Campos'),  
(4, 'JULIETA PALACIOS', '10501160-1', 49, 'CONSULTORIO', 'SALAMANCA',  
'Gloria Campos'),  
(5, 'GRACE ROJAS', '15047780-8', 33, 'CONSULTORIO', 'LLIMPO', 'Gloria  
Campos'),  
(6, 'MABEL PALACIOS', '10501168-7', 51, 'CONSULTORIO', 'SALAMANCA',  
'Doctor');  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

### **B.3.2 Conexión entre la base de datos y la interfaz gráfica (conexión\_bd.php)**

Este programa permite extraer todos los datos almacenados en la base de datos, con el fin de mostrarlos en pantalla una vez que se requieran por el usuario que está utilizando la interfaz gráfica del sistema.

```
<?php  
$bd_servidor = "localhost";  
$bd_usuario = "esteban";  
$bd_clave = "ehealth4321";  
$base_de_datos = "ehealth_bd";  
$conexion = mysql_connect($bd_servidor, $bd_usuario, $bd_clave) or  
die(mysql_error());  
mysql_select_db($base_de_datos, $conexion) or die(mysql_error());  
?>
```