

INFORME FINAL DE INVESTIGACIÓN

**DESARROLLO DE UN SISTEMA BIOMÉDICO
PARA MONITOREO LOCAL Y REMOTO DE
SIGNOS VITALES DE PACIENTES,
UTILIZANDO TECNOLOGÍA DE INTERNET
DE LAS COSAS IoT**

En asocio con el Hospital Santa Teresa
de Zacatecoluca

DOCENTE INVESTIGADOR PRINCIPAL:
LIC. MANUEL DE JESÚS GÁMEZ LÓPEZ

CARRERA TÉCNICO EN SISTEMAS INFORMÁTICOS
ITCA-FEPADÉ CENTRO REGIONAL MEGATEC ZACATECOLUCA

ENERO 2020

INFORME FINAL DE INVESTIGACIÓN

**DESARROLLO DE UN SISTEMA BIOMÉDICO
PARA MONITOREO LOCAL Y REMOTO DE
SIGNOS VITALES DE PACIENTES,
UTILIZANDO TECNOLOGÍA DE INTERNET
DE LAS COSAS IoT**

En asocio con el Hospital Santa Teresa
de Zacatecoluca

DOCENTE INVESTIGADOR PRINCIPAL:
LIC. MANUEL DE JESÚS GÁMEZ LÓPEZ

CARRERA TÉCNICO EN SISTEMAS INFORMÁTICOS
ITCA-FEPADÉ CENTRO REGIONAL MEGATEC ZACATECOLUCA

ENERO 2020

Rectora

Licda. Elsy Escolar SantoDomingo

Vicerrector Académico

Ing. Carlos Alberto Arriola Martínez

Vicerrectora Técnica Administrativa

Inga. Frineé Violeta Castillo

**Director de Investigación
y Proyección Social**

Ing. Mario W. Montes Arias

**Dirección de Investigación
y Proyección Social**

Ing. David Emmanuel Ágreda Trujillo

Inga. Ingrid Janeth Ulloa de Posada

Sra. Edith Aracely Cardoza de González

**Director Centro Regional
MEGATEC Zacatecoluca**

Ing. Christian Antonio Guevara Orantes

616.024 028 5

G192d

Gómez López, Manuel de Jesús, 1984 -

slv

Desarrollo de un sistema biomédico para monitoreo local y remoto de signos vitales de pacientes, utilizando tecnología de Internet de las Cosas IoT [recurso electrónico] : en asocio con el Hospital Santa Teresa de Zacatecoluca / Manuel de Jesús Gómez López. -- 1ª ed. -- Santa Tecla, Libertad, El Salv. : ITCA Editores, 2020.

1 recurso electrónico (72 p. : il. col. ; 28 cm.)

Datos electrónicos (1 archivo : pdf, 5.7 mb). -

<https://www.itca.edu.sv/produccion-academica/>

ISBN 978-99961-39-49-9 (E-Book, pdf)

ISBN 978-99961-39-35-2 (Impreso)

1. Signos vitales – Equipo – Automatización. 2. Medicina Popular – Aparatos e instrumentos. 3. Internet de las Cosas. I. Título.

Autor

Lic. Manuel de Jesús Gómez López

Tiraje: 13 ejemplares

Año 2020

Este documento técnico es una publicación de la Escuela Especializada en Ingeniería ITCA-FEPADE; tiene el propósito de difundir la Ciencia, la Tecnología y la Innovación CTI, entre la comunidad académica, el sector empresarial y la sociedad, como un aporte al desarrollo del país. Para referirse al contenido debe citar el nombre del autor y el título del documento. El contenido de este Informe es responsabilidad de los autores.



Atribución-No Comercial
Compartir Igual
4.0 Internacional

Esta obra está bajo una licencia Creative Commons. No se permite el uso comercial de la obra original ni de las posibles obras derivadas, cuya distribución debe hacerse mediante una licencia igual que la sujeta a la obra original.

Escuela Especializada en Ingeniería ITCA-FEPADE

Km 11.5 carretera a Santa Tecla, La Libertad, El Salvador, Centro América

Sitio Web: www.itca.edu.sv

TEL: (503)2132-7423

CONTENIDO

1.	INTRODUCCIÓN	4
2.	PLANTEAMIENTO DEL PROBLEMA.....	5
2.1.	DEFINICIÓN DEL PROBLEMA	5
2.2.	ANTECEDENTES / ESTADO DE LA TÉCNICA	5
2.3.	JUSTIFICACIÓN.....	7
3.	OBJETIVOS	8
3.1.	OBJETIVO GENERAL	8
3.2.	OBJETIVOS ESPECÍFICOS	8
4.	HIPÓTESIS	8
5.	MARCO TEÓRICO.....	8
5.1.	CONCEPTOS Y DEFINICIONES.....	8
6.	METODOLOGÍA DE INVESTIGACIÓN	14
6.1.	MATRIZ OPERACIONAL DE LA METODOLOGÍA	15
7.	RESULTADOS.....	18
8.	CONCLUSIONES	30
9.	RECOMENDACIONES	31
10.	GLOSARIO	32
11.	REFERENCIAS BIBLIOGRÁFICAS.....	33
12.	ANEXOS	34

1. INTRODUCCIÓN

El auge de las tecnologías basadas en el concepto de IoT (Internet of Things), ofrece diversidad de uso, uno de estos es el hecho de mejorar el servicio de atención sanitaria con el objetivo de poder prevenir situaciones o estados críticos de salud en pacientes y así salvar vidas. Al recopilar datos de dispositivos de monitorización doméstico y/o hospitalarios, y con ello consultar información de pacientes y realizar diagnósticos en tiempo real desde cualquier ubicación por parte de un especialista o un sistema inteligente (Big Data), puede mejorar la calidad del actual servicio brindado por el sistema de atención sanitario. Gracias al avance de la tecnología en diferentes ramas de las ciencias, tal como la medicina (Instrumentación Biomédica y/o dispositivos biométricos), se ha logrado desarrollar instrumentos electrónicos que facilitan el estudio de señales del cuerpo humano para comprender su comportamiento y/o patología presente. Además, este tipo de proyecto brinda una experiencia nueva a los pacientes y encargados de la atención sanitaria, adentrándolos al globalizado mundo de la transformación digital en el sector salud respecto a la atención sanitaria o médica.

La plataforma para e-Health desarrollada por Cooking Hacks [1], es llamada MySignals HW 2.0 [2]. creada con el objetivo de medir datos de sensores biométricos para fines de experimentación, funciona en conjunto con la plataforma Arduino, la cual es una placa de desarrollo de hardware libre aplicada en proyectos multidisciplinarios. Este proyecto se enfoca en el desarrollo de un prototipo de dispositivo electrónico portátil para realizar la captura de información biométrica a través de sensores, almacenar datos y visualizarlos desde diferentes dispositivos inteligentes. Su ventaja es que proporciona bondades que permiten mejorar los servicios de telemedicina en el sistema de salud.

En este proyecto se desarrolló el prototipo y pruebas de verificación del funcionamiento necesarias en un equipo médico, con el fin de obtener información confiable de dicho dispositivo. Se enfocó la investigación en el área de la tecnología aplicada en el sector de la salud, desarrollando un sistema de monitoreo local y remoto de signos vitales de pacientes utilizando como base la tecnología IoT y dispositivos inteligentes Smartphone. Para ello se diseñó una App Android, un dispositivo electrónico biomédico a partir de la tarjeta electrónica MySignals HW [2], así como la integración y programación de sensores biométricos con el controlador y periféricos de visualización de la información (Pantalla táctil TFT 2.4", TJCTM24024-SPI), el cual se dota con la capacidad necesaria de comunicación para que, a través de la red se envíen los datos obtenidos como resultado del proceso de lectura de los sensores. Estos resultados se envían de forma instantánea a la base de datos en el servidor IoT, lo que permite saber desde cualquier lugar, las lecturas medicas tomadas del paciente por medio de dispositivos inteligentes a través de la App o portal web. Los sensores utilizados y probados en esta investigación son: Oximetría (SpO2 y Pulso), Temperatura corporal, frecuencia respiratoria y tensión o presión arterial.

Una vez realizadas pruebas con personal del Hospital Santa Teresa de Zacatecoluca, se pudo establecer que el prototipo arroja valores muy cercanos en comparación con los de un equipo especializado. Proporcionando este un grado de confiabilidad del 95% para las lecturas de temperatura corporal, saturación parcial del oxígeno y frecuencia cardiaca. Para la variable frecuencia respiratoria no se pudo efectuar comparaciones debido a que dicha medición la toman basada en otro método a través de la conexión de electrodos, siendo un sensor diferente al de esta investigación.

Este proyecto fue desarrollado por la carrera de Técnico en Sistemas Informáticos de ITCA-FEPADE Centro Regional MEGATEC Zacatecoluca.

2. PLANTEAMIENTO DEL PROBLEMA

2.1. DEFINICIÓN DEL PROBLEMA

La toma de señales bioeléctricas es un método importante de prevenir enfermedades y monitorear el avance de un paciente. La realización de toma de estas constantes vitales, normalmente implica el uso de equipos especializados y tecnologías de difícil aplicación. Es por lo que a través de la Ingeniería Electrónica y los avances de la telemedicina se han venido desarrollando distintas plataformas electrónicas multifuncionales que ayudan a la toma de estas señales bioeléctricas de una forma eficiente y eficaz. Lo anterior Se hace por medio del uso de sistemas compuestos por sensores y equipos de parametrización. El monitoreo de señales bioeléctricas en el país es llevado a cabo en los centros hospitalarios con el fin de revisar y llevar control de signos vitales de pacientes.

Se identifica que el actual equipamiento biomédico especializado con el que se cuenta en los centros hospitalarios públicos para brindar atención sanitaria (medición de los signos vitales) de pacientes a nivel nacional, carecen de la funcionalidad tecnológica y fundamental de tener interconectados a la red global de Internet los datos clínicos generados en el proceso de brindar una asistencia sanitaria por medio de los aparatos especializados. Esto evita o bloquea a los responsables (especialistas) a no poder realizar un mejor trabajo de monitoreo y supervisión a los pacientes, debido a la falta de funcionalidades en los equipos utilizados y específicamente de comunicación e integración con otras plataformas para compartir dicha información del estado y eventos clínicos ocurridos en un paciente. Por lo que nace la gran e importante necesidad de implementar estrategias que no solo permitan el uso de las tecnologías en el sector salud, sino también, permitan optimizar el acceso y manejo de datos de suma relevancia, debido a que gracias al acceso oportuno de esta información se puede llegar al relevante hecho de salvar vidas humanas, además de poder llevar un mejor control digitalizado de la información. Este hecho obliga al/los especialistas en atención sanitaria estar frente al paciente y monitor (equipo biomédico especializado) para tener acceso a dicha información. Sin embargo, no siempre el especialista por diversas razones va a poder presenciar cualquier ocurrencia en el paciente cuando no esté presente.

En conclusión, no se cuenta con la ventaja tecnológica de poder consultar la información de manera remota, ni de informar, alertar o notificar al especialista de dichos estados clínicos, no importando la localidad de este; en el momento preciso para la toma de decisiones oportunas. Pudiendo de esta manera prevenir lamentables sucesos y a la vez de brindarle una mayor seguridad y confianza en el servicio brindado a los pacientes.

2.2. ANTECEDENTES / ESTADO DE LA TÉCNICA

La innovación es el motor de los avances tecnológicos. Esta afirmación se base en investigaciones realizadas por J. Schum Peter quien resalta que “el desarrollo económico está motivado por la innovación, por medio de un proceso dinámico en el cual nuevas tecnologías sustituyen a las antiguas”.

La industria de la tecnología en salud ha comprendido este fenómeno, en efecto, los dispositivos médicos se han convertido en un negocio que presenta dinámicas de crecimiento notorias a nivel mundial. El área de la tecnología biomédica ha evolucionado de forma acelerada durante las últimas décadas, situación que ha mejorado la calidad de los servicios de salud, al ofrecer diferentes soluciones y tratamientos para las enfermedades, transformando a las instituciones de salud en espacios sofisticados en tecnología.

La convergencia de disciplinas como la nanotecnología, la biología, los sistemas de información, el Internet de las Cosas y la Inteligencia Artificial continuarán impulsando aún más la innovación de estos sistemas convirtiéndolos en una poderosa herramienta para el desarrollo de la salud humana.

Casi tres cuartas partes de los líderes en atención sanitaria que han adoptado IoT consideran que su principal ventaja será la posibilidad de supervisar y controlar variables de salud provenientes de dispositivos médicos desde cualquier ubicación, en el momento preciso de entornos sanitarios. Es posible con las nuevas herramientas tecnológicas a disposición, crearse un entorno más seguro y eficaz de supervisión y gestión de datos médicos de pacientes. Con una única aplicación en un dispositivo móvil los pacientes y el personal pueden gestionar de forma segura los datos.

A raíz de la investigación realizada dentro del marco de la biomédica y la tecnología se identificó que en el país (El Salvador, centros hospitalarios públicos) no se cuenta con servicios/espacios especializados en atención sanitaria a pacientes y específicamente para gestión de los datos clínicos de manera óptima, pudiéndose incorporar la tecnología en sus procedimientos actuales o formas de brindar dicho servicio, siendo este un lamentable hecho y necesidad inherente en dicho sector.

Si bien, la biomédica ha tenido un desarrollo impresionante a nivel mundial, en el país a pesar de las propuestas actuales. Falta potencializar mucho la investigación técnica científica que busque beneficios en costos y procedimientos seguros, eficientes y rápidos, tanto en diagnóstico como en tratamiento y rehabilitación, según la propia necesidad y de los avances de la aplicación de la medicina en el país y región, que va vinculada a la gran demanda de salud que afectan fuertemente indicadores clínicos, y para ello no solo con querer desde la cátedra es suficiente si no existe interés del sector privado y gubernamental.

A nivel mundial, se tienen fuertes aportes tecnológicos basados en el enfoque de esta investigación. A continuación, se hace mención de algunas de éstas:

- **Monitor de glucosa FreeStyle Libre.** Desarrollado por Abbott, el sistema FreeStyle Libre es un dispositivo de monitoreo continuo de glucosa a través de un pequeño sensor que se aplica en la parte posterior del brazo. El dispositivo está indicado para reemplazar las pruebas de glucosa en sangre y detectar tendencias y patrones de seguimiento que ayudan en la detección de episodios de hiperglucemia e hipoglucemia, facilitando los ajustes de terapia aguda en personas a largo plazo con diabetes.
- **Sistema de visualización SpyGlass.** El SpyGlass, de Boston Scientific, es un sistema de visualización directa que se utiliza para las aplicaciones diagnósticas y terapéuticas como cáncer del sistema biliar, cáncer de páncreas, cáncer de conductos, cáncer de vesícula biliar, cálculos de esclerosis, pancreatitis entre otras patologías del páncreas. Su diseño permite optimizar la eficiencia de los procedimientos y la productividad debido a su facilidad de configuración y uso además de la alta calidad de imagen que ofrece.
- **Dispositivos vestibles o llevables (Wearables).** Hace referencia al conjunto de aparatos y dispositivos electrónicos que se incorporan en alguna parte de nuestro cuerpo interactuando de forma continua con el usuario y con otros dispositivos con la finalidad de realizar alguna función concreta, relojes inteligentes o Smartwatch, zapatillas de deportes con GPS incorporado y pulseras que controlan nuestro estado de salud son ejemplos entre otros muchos de este género tecnológico presente en nuestras vidas.

2.3. JUSTIFICACIÓN

Gracias a los avances de las telecomunicaciones, sistemas de información y las tecnologías e-Health, se han generado escenarios para el nuevo desarrollo de tecnologías en el área de la salud y la medicina, específicamente en la telemedicina (tecnologías de salud o la prestación de servicios médicos a distancia, implementando tecnologías de la información), la cual permite expandir la diversidad de sus servicios y aumento de cobertura regional.

Sin embargo. Todavía existen barreras en infraestructuras tecnológicas centradas en los costos exagerados de equipos especializados y en la no definición de arquitecturas telemáticas abiertas y modulares de precisión en su lectura, en especial en aquellas centradas en el monitoreo remoto de variables clínicas de pacientes y enfermedades crónicas.

Actualmente en el país, hay muchos que no tienen la oportunidad de asistir a clínicas o centros médicos para tener un monitoreo constante de signos vitales, ya sea por razones económicas, por conformismo o confianza en mejorar sin atención adecuada, por el acceso a los centros hospitalarios o de salud, etc. Así mismo, hoy en día existen centros médicos que no pueden cubrir la alta demanda de pacientes y sus respectivos cuidados. Algunas personas van por urgencias que requieren de una hospitalización y cuidado adecuado, mientras que otras personas no necesitan permanecer constantemente en el hospital o centro clínico. Sin embargo, es necesario estar supervisando sus signos vitales, no importando su geolocalización, siendo este último caso uno de los posibles escenarios donde se puede recurrir a este prototipo, Por otro lado, los aparatos médicos son muy caros, no están al alcance de las personas. Por tales razones se desarrolla el prototipo electrónico utilizando sensores biométricos existentes en el mercado para la medición de signos vitales de pacientes, el cual por un lado será mucho más accesible económicamente y de fácil uso, para el paciente y el especialista.

Se crea el dispositivo con la idea que pudiese adquirirse y manejarse fácilmente. La idea no es sustituir el equipamiento actual hospitalario. Solo se busca crear un dispositivo electrónico biomédico o herramienta básica para fabricar equipos más cómodos que facilitasen el control y supervisión de los estados clínicos de un paciente desde cualquier ubicación (local [Hogar o centro de salud] o remota).

En la investigación de campo realizada para la búsqueda del asocio colaborativo, se identificó que el actual equipamiento biomédico en centros hospitalarios públicos no se cuenta con un sistema de comunicaciones como el que se realiza e integra al prototipo electrónico de esta investigación y específicamente en el monitoreo remoto de estados o eventos clínicos de pacientes. Cabiendo la posibilidad de desarrollar futuras investigaciones aplicadas en equipo biomédico especializado, existente en la red nacional de salud, con el fin de escalar el nivel de monitoreo y la calidad de servicio brindado, a través de la implementación de la comunicación e integración con otras plataformas. IoT ofrece una experiencia de atención y supervisión de pacientes más sencilla y eficiente, permitiendo que el personal sanitario haga mejor su trabajo, además de hacer posible que sea el mismo paciente el protagonista en llevar un autocuidado de su salud y así poder prevenir (basándose en tablas medicas estandarizadas), también se incorpora la opción de compartir dicha información con un especialista para un diagnostico más confiable.

Con el desarrollo de esta investigación se espera obtener beneficios a nivel social y económico en cuanto a la creación y optimización de servicios sanitarios en telemedicina móvil.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Desarrollar un sistema electrónico biomédico para la medición de signos vitales de pacientes aplicando la tecnología de Internet de las Cosas IoT.

3.2. OBJETIVOS ESPECÍFICOS

- Diseñar un prototipo electrónico con la integración de los diferentes componentes electrónicos del sistema físico.
- Desarrollar un sistema de comunicación para el monitoreo de las variables fisiológicas.
- Desarrollar una App (Android) para el monitoreo de signos vitales de pacientes.

4. HIPÓTESIS

¿Cómo optimizar el servicio de atención sanitaria por medio de la medición de variables médicas (signos vitales) de pacientes aplicando la tecnología de Internet de las Cosas IoT?

5. MARCO TEÓRICO

5.1. CONCEPTOS Y DEFINICIONES

Internet de las cosas (en inglés, Internet of Things, abreviado IoT), es un concepto que se refiere a la interconexión digital de objetos cotidianos con Internet. Alternativamente, Internet de las Cosas es la conexión de Internet con “cosas u objetos” que personas. También se suele conocer como Internet de todas las cosas o Internet en las Cosas.

IoT es una red de objetos digitalizados y dotados con características particulares de cómputo, comunicación y sensorial, los cuales están interconectados a la nube (Internet) y/o entre ellos realizando una tarea específica en el proceso, proporcionando información de interés al centro de datos, la cual normalmente es enviada y recibida en tiempo real para la toma de decisiones y acciones oportunas por humanos o por un sistema M2M (Machine to Machine).

Sistemas e-Health, Salud electrónica o e-Health es el término con el que se define al conjunto de Tecnologías de la Información y la Comunicación (TIC) que son usadas en el ámbito de medicina para una mejor contribución.

En los últimos años ha cogido gran fuerza gracias al avance de las tecnologías que se aplican en el tema de salud pública. e-Health es definida como “el uso de información digital, transmitida, almacenada u obtenida electrónicamente para el apoyo al cuidado de la salud tanto a nivel local como a distancia” y se sustenta en el uso de las Tecnologías de la Información y las Comunicaciones (TIC) en aspectos que van desde la gestión de las organizaciones, el acceso a información relevante, el seguimiento del paciente, o incluso la posibilidad de realizar diagnósticos alternativos en enfermedades.

Arduino. Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. Es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar.

Arduino puede medir entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el IDE de Arduino.

Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador. Las placas se pueden ensamblar a mano o encargarlas pre-ensambladas; el software se puede descargar gratuitamente.

Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que se tiene libertad de adaptarlas a las necesidades.

Microcontrolador. Un microcontrolador (abreviado μC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.

Sistemas de Comunicación.

Bluetooth, es una especificación tecnológica para redes inalámbricas que permite la transmisión de voz y datos entre distintos dispositivos mediante una radiofrecuencia segura (2,4 GHz). Esta tecnología, por lo tanto, permite las comunicaciones sin cables ni conectores y la posibilidad de crear redes inalámbricas domésticas para sincronizar y compartir la información que se encuentra almacenada en diversos equipos.

Wifi, es una tecnología de comunicación inalámbrica que permite conectar a Internet equipos electrónicos, como computadoras, tablets, smartphones o celulares, etc., mediante el uso de radiofrecuencias o infrarrojos para la transmisión de la información.

Internet, Red informática de nivel mundial que utiliza la línea telefónica para transmitir la información.

Tarjeta Electrónica e-Health o e-Salud.

MySignals, es una plataforma de desarrollo para dispositivos médicos y aplicaciones de e-Health. Puede usarse MySignals para desarrollar la plataforma web de e-Health o incluso agregar sensores propios para construir nuevos dispositivos médicos.

Sensor, es un objeto capaz de detectar magnitudes físicas, bioeléctricas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas.

Android, es un sistema operativo móvil que se basa en una versión modificada de Linux. En 2005, como parte de su estrategia para entrar en el mundo móvil, Google compró Android y se hizo cargo de su trabajo de desarrollo (así como de su equipo).

Google quería que Android fuera abierto y libre; de ahí que la mayoría del código Android se pusiera disponible bajo licencia Apache de código abierto, lo que significa que cualquiera que quiera utilizar Android puede hacerlo al descargar su código fuente completo.

La principal ventaja de adoptar Android es que ofrece un enfoque unificado para el desarrollo de aplicaciones. Los desarrolladores solamente necesitan desarrollar para Android, y sus aplicaciones

deberían poderse ejecutar en numerosos dispositivos, siempre y cuando los dispositivos utilicen este sistema operativo.

Gradle, es una herramienta para automatizar la construcción de proyectos, es decir la compilación o el testing por ejemplo, además gestiona las dependencias. Está basada en Groovy, un lenguaje de programación muy parecido a Java y está incorporada en Android Studio.

Base de datos, es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico. Las bases de datos tradicionales se organizan por campos, registros y archivos.

MySQL es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL, y todo el mundo puede contribuir para incluir elementos, arreglar problemas, realizar mejoras o sugerir optimizaciones.

MySQL es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2.

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos.

SQLite, es un motor de bases de datos de código abierto, transaccional, que no necesita configuración ni requiere un servidor y se caracteriza por mantener el almacenamiento de información de forma sencilla.

Es una tecnología cómoda para los dispositivos móviles a pesar de que tiene ciertas limitaciones en determinadas operaciones. Se ha utilizado este motor para crear y testear la base de datos de la aplicación Android.

ORMLite (Object Relational Mapping Lite) es una librería open source que permite mapear objetos Java y persistirlos en una base de datos SQL. Utiliza clases abstractas para el acceso a los datos (DAO) y soporta SQLite. Dado que en este proyecto se usa esta base de datos se ha utilizado esta librería para gestionar las entidades, así como los datos de las aplicaciones.

DB Browser for SQLite, es una herramienta open source con la que se puede gestionar archivos de bases de datos SQLite disponible para Windows, Mac OS X, Linux y FreeBSD. Gracias a esta herramienta que fue posible inspeccionar el contenido de la base de datos que generaba nuestra aplicación, dentro de los emuladores. Cuando se importan y se abren con esta herramienta, aparecen claramente las distintas tablas con sus columnas con respectivos registros de información, lo cual permite comprobar el correcto o incorrecto manejo de los datos.

App, es una aplicación de software que se instala en dispositivos móviles o tablets para ayudar al usuario en una labor concreta, a diferencia de una WebApp que no es instalable. El objetivo de una app es facilitar la consecución de una tarea determinada o asistirnos en operaciones y gestiones del día a día. Para el desarrollo de la App, se utilizó el IDE de Android Studio 3.5.3.

Archivo APK, fichero que se genera en el proceso de compilación del código JAVA y clases de Android Studio, el cual se utilizará para la instalación en dispositivos móviles o smartphone con sistema operativo Android.

El instalador generado (apk) versión release se ha probado e instalado en dispositivos con sistema operativo Android 4.4 (KitKat) en adelante hasta la versión 8 (Oreo), en celular y Tablet.

1. Tecnologías utilizadas para el desarrollo del Software del Sistema.

(Plataforma Web, Web Service, Base de datos y App)



Fig. 1. Herramientas de desarrollo y diseño.

- ✚ Editores de texto para codificación: Visual Studio Code y Atom.
- ✚ Software de control de versiones: Git.
- ✚ IDE de desarrollo para el código fuente del controlador del sistema: Arduino.
- ✚ Framework de diseño para portal web y Dashboard: Bootstrap, Materialize design, HTML5 y CSS.
- ✚ IDE de desarrollo de la aplicación móvil para Smartphone: Android Studio.
- ✚ Lenguaje de programación para la codificación y funcionalidad de la App: JAVA.
- ✚ Lenguaje de programación para la codificación y funcionalidad del portal web y web service: PHP, JavaScript y Ajax.
- ✚ Base de datos para almacenamiento de información del sistema: SQLite y MySQL.
- ✚ Software para gestión de la base de datos XAMPP CONTROL PANEL V3.2.2.

2. Hardware del Sistema.



Fig. 2. Controlador del sistema.

En la Fig. 2, se muestra el controlador utilizado para el prototipo. Arduino Mega 2560 es una placa de microcontrolador basada en el ATmega2560. Tiene 54 pines de entrada / salida digital (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP, y un botón de reinicio.

Tabla 1. Especificaciones Técnicas.

Microcontrolador	ATmega2560
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Pines de E / S digitales	54 (de los cuales 15 proporcionan salida PWM)
Pines de entrada analógica	dieciséis
Corriente CC por pin de E / S	20 mA
Corriente CC para Pin de 3.3V	50 mA
Memoria flash	256 KB de los cuales 8 KB utilizados por el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Longitud	101,52 mm
Anchura	53,3 mm
Peso	37 g

El controlador del sistema es la tarjeta electrónica responsable de almacenar y ejecutar el código fuente que hace posible:

- Establecer comunicación bidireccional por canal UART, para obtener y ejecutar acciones en base a la información médica procesada por la tarjeta MySignals (HW Development Platform V2.0).
- Establecer comunicación bidireccional por canal UART, vía Bluetooth para enviar los datos obtenidos de los sensores biométricos a la aplicación móvil, y a la vez para recibir las ordenes o comandos de control desde la App para setear el sistema en un modo de funcionalidad o medición en específico.
- Establecer comunicación unidireccional por canal UART con un ordenador utilizando software de terceros para visualización de la información procesada por el sistema.

3. MySignals HW V2.0.

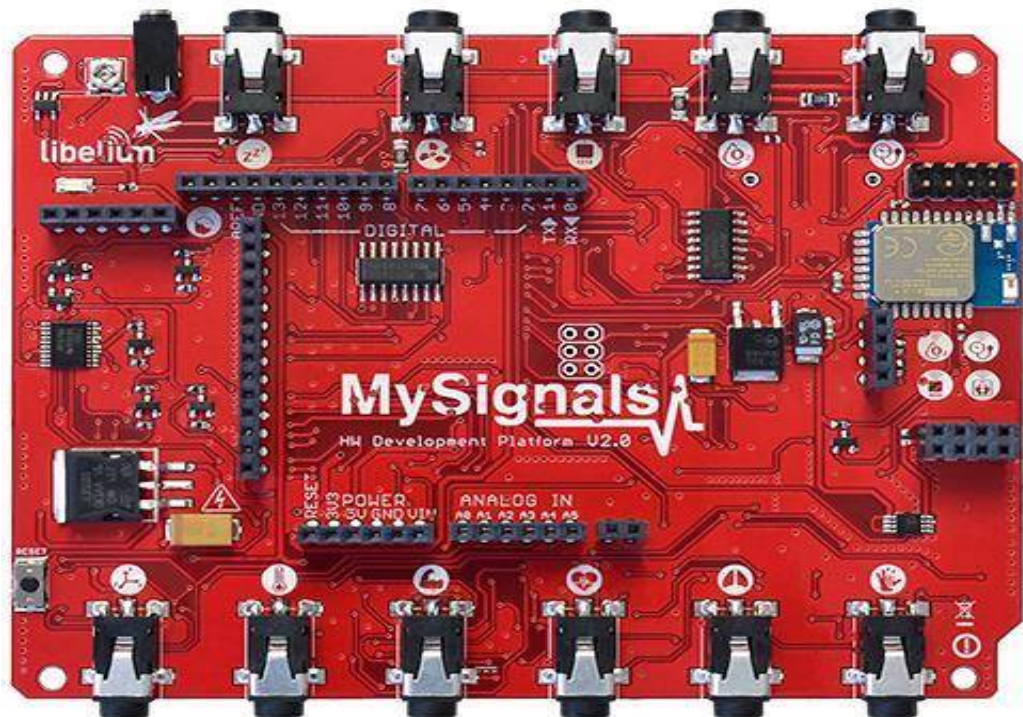


Fig. 3. Tarjeta MySignals HW 2.0.

En la Fig. 3, se muestra la tarjeta electrónica que interactúa con el controlador y a la vez la responsable de procesar y acondicionar las señales biométricas del sistema. MySignals HW Complete Kit [3] incluye múltiples sensores que pueden usarse para monitorear muchos parámetros biométricos. Para esta investigación se incluye la medición de las siguientes variables biométricas: Oximetría (Saturación parcial del oxígeno y frecuencia cardíaca), frecuencia respiratoria, temperatura corporal y tensión o presión arterial. Los datos se pueden visualizar en una tableta o teléfono inteligente a través de la aplicación o App Android y con una computadora a través de un navegador web desde el portal web.

Tabla 2. Características Generales.

	My Signals HW
Arquitectura	Compatible con Arduino
Memoria RAM	2k, 8k
Procesador	ATmega 328 (Arduino UNO), ATmega 2560
Memoria Flash	32k, 256k
UART	1 (Multiplexado)
Pantalla	TFT (Gráficos básicos)
Sensores	16
Sensores cableados	11
Sensores inalámbricos	16
Lecturas concurrentes de sensores	De un grupo de sensores (Analógico, UART, BLE) a una interfaz (TFT, BLE, WiFi)
Radios en tarjeta	BLE, WiFi
Radios Extra que soporta	BT, ZigBee, 4G / 3G / GPRS

Dentro de las posibilidades de manejo de información biométrica por sensores que están disponibles en el mercado y que son compatibles con la tarjeta se tienen: Posición del cuerpo, temperatura corporal, electromiografía, electrocardiografía, flujo de aire, respuesta galvánica de la piel, presión sanguínea, Pulsioxímetro, glucómetro, espirómetro, ronquido, escala (BLE), presión sanguínea (BLE), pulsioxímetro (BLE), glucómetro (BLE), electroencefalografía.

El Hardware de MySignals con Arduino Mega puede ser alimentado por la PC o por una fuente de alimentación externa (12V – 2A), siendo la más recomendable de utilizar la última opción (Fuente externa).

6. METODOLOGÍA DE INVESTIGACIÓN

El proyecto comprende el desarrollo de un sistema de monitoreo local y remoto de signos vitales de pacientes utilizando la tecnología de IoT (Internet of Things), para ello se diseña y desarrolla una App (Aplicación para Smartphone), la cual permitirá el control y la monitorización de la información clínica de pacientes, además, se desarrolla el dispositivo electrónico biomédico a partir de la tarjeta electrónica MySignals HW y sensores biométricos existentes en el mercado, finalmente se realizará la integración y programación de sensores biométricos con el controlador y periféricos de visualización de la información (Pantalla GLCD); el cual se dotará con la capacidad necesaria de comunicación para que, a través de la red se envíen los datos obtenidos como resultado del proceso de la/las lecturas de los sensores biométricos.

Para ello y por ser una metodología de investigación flexible y adaptable al escenario de este trabajo se optó por desarrollar las diferentes fases del proyecto, bajo la metodología de Gestión de proyectos AGILE, la cual es una metodología de gestión de proyectos ampliamente usada en el sector IT (Information

Technology) y proyectos de organización empresariales, que tiene como principal virtud la flexibilidad y capacidad de modificar el producto a lo largo del proyecto, ya que estos se van usando al mismo tiempo que se desarrollan.

Esta metodología se basa en dividir el proyecto en fases (Sprints), el resultado de las cuales es un producto con una serie de funcionalidades que ya permiten que este sea usado. Estas fases se terminan hasta haber conseguido el total de las funcionalidades definidas para el producto.

De forma resumida las fases se componen de lo siguiente:

Inicio. Se escoge del total de objetivos del producto aquellos que serán implementados en el Sprint, debiendo ser capaces de generar un producto funcional. En base a estos objetivos se define la duración del Sprint (entre una semana y un mes), y las tareas que lo componen.

Desarrollo del Sprint. El equipo del proyecto planifica y ejecuta las tareas, las cuales se van supervisando en reuniones diarias donde se miran las tareas ejecutadas, en curso, y pendientes, así como posibles impedimentos y restricciones.

Cierre. Al final del Sprint se revisa que se hayan completado las tareas y objetivos definidos al inicio mediante la presentación de un producto funcional. El ensayo de este producto por parte del cliente puede hacer variar los objetivos del proyecto o su prioridad.

La posibilidad de tener un producto funcional y utilizable al final de cada Sprint permite ir ajustando los objetivos del proyecto, y por tanto asegurar mejor que el producto final cumplirá con las expectativas del usuario. A parte, la comercialización de estos Sprints permite avanzar los ingresos generados por el proyecto, mejorando su rentabilidad.

6.1. MATRIZ OPERACIONAL DE LA METODOLOGÍA

Tabla 3. Matriz metodológica.

OBJETIVOS ESPECÍFICOS	ACTIVIDADES A EJECUTAR	RESULTADOS ESPERADOS	MATERIALES
Realizar la planificación de la investigación.	<ul style="list-style-type: none"> • Investigación bibliográfica. • Investigación acerca del estado de la técnica. • Investigación de campo. • Investigación de materiales para el proyecto. • Redacción del Anteproyecto. • Presentación de anteproyecto a sede central. 	Aprobación de Anteproyecto.	Computadora o Smartphone.

OBJETIVOS ESPECÍFICOS	ACTIVIDADES A EJECUTAR	RESULTADOS ESPERADOS	MATERIALES
Desarrollar la aplicación móvil para dispositivos Smartphone Android.	<ul style="list-style-type: none"> • Instalación y configuración del IDE de desarrollo de aplicaciones móviles (App). • Uso del Framework material Design para el diseño de la App. • Diseño de un Dashboard para visualización de la información clínica de pacientes. • Creación de la codificación para el manejo de Bluetooth del dispositivo móvil. • Creación de la codificación para la monitorización local y remota de variables clínicas de pacientes a través de la App. 	Aplicación móvil para dispositivos Smartphone Android.	Computadora, Smartphone Android, cable USB, Bluetooth, Arduino.
Diseñar la base de datos del sistema.	<ul style="list-style-type: none"> • Instalación de XAMPP CONTROL PANEL para obtener el gestor de base de datos MYSQL. • Instalación de MySQL Workbench. • Uso de MySQL Workbench para el diseño de la base de datos. • Generar el script SQL de la base de datos. 	Diseño de la base de datos del sistema.	Computadora y software SGBD y de diseño.
Desarrollar la comunicación para la interconexión e interoperabilidad de las diferentes partes que conforman el sistema.	<ul style="list-style-type: none"> • Creación de dominio en servidor online ITCA u otro. • Creación de un web service o API. • Realizar pruebas de funcionalidad en el servidor web. • Uso del lenguaje de programación PHP. • Creación de la codificación para la interoperabilidad de la App, web service y la base de datos. 	Web Service o API.	Computadora, dispositivo Android, cable USB, Bluetooth, Dominio online (preferiblemente) o servidor web local para previas pruebas.

OBJETIVOS ESPECÍFICOS	ACTIVIDADES A EJECUTAR	RESULTADOS ESPERADOS	MATERIALES
Comprar los materiales del proyecto.	<ul style="list-style-type: none"> • Generación de lista de materiales con especificaciones técnicas. • Investigación acerca de posibles proveedores. • Visitas a proveedores. • Compra de materiales del proyecto. 	Materiales.	Computadora.
Desarrollar el dispositivo electrónico biomédico.	<ul style="list-style-type: none"> • Instalación y uso del IDE de Arduino para programación de los componentes electrónicos del sistema. • Descarga de bibliotecas para el kit de sensores fisiológicos y su integración con el controlador Arduino. • Codificación de Arduino para la comunicación con la tarjeta electrónica MySignal HW y sensores biométricos. • Codificación del escudo de comunicación con Arduino, para conexión con la red (Internet), web service y la base de datos en el servidor online. • Verificación de funcionalidad. • Depuración de errores. 	Prototipo electrónico biomédico sin chasis.	Computadora, Tarjeta electrónica MySignals, sensores biométricos, GLCD, Controlador, Módulo Bluetooth, cables header, cable USB, entre otros.
Diseñar el chasis para montaje del circuito final.	<ul style="list-style-type: none"> • Instalación de Software CAD. • Uso de software CAD. 	Chasis.	Computadora, impresora.
Ensamblar los componentes electrónicos en el chasis.	<ul style="list-style-type: none"> • Montaje de todos los componentes electrónicos del sistema en su respectivo chasis. • Pruebas de funcionalidad del producto final. 	Prototipo electrónico biomédico con chasis.	Computadora, Tarjeta electrónica MySignals HW, Sensores biométricos, GLCD, Smartphone Android, App, Otros.

OBJETIVOS ESPECÍFICOS	ACTIVIDADES A EJECUTAR	RESULTADOS ESPERADOS	MATERIALES
Generar la tabla de información con parámetros médicos estándares de las variables del sistema.	<ul style="list-style-type: none"> Programación de reunión con especialista de atención sanitaria en Hospital Santa Teresa, Zacatecoluca, la Paz. Visita a Hospital Santa Teresa, Zacatecoluca, la Paz. Dialogo respecto a datos ya establecidos o estándares de variables clínicas de pacientes con especialista. 	Tabla de parámetros médicos estandarizados.	Computadora o Smartphone.
Realizar pruebas de funcionamiento del dispositivo electrónico biomédico, para determinar precisión y confiabilidad de las mediciones clínicas de pacientes.	<ul style="list-style-type: none"> Visita a Hospital Santa Teresa, Zacatecoluca, la Paz. Puesta en marcha: pruebas del dispositivo electrónico biomédico en pacientes y supervisado por el especialista de atención sanitaria. 	Determinación de los resultados finales.	Computadora, Prototipo Biomédico, Smartphone Android, App, paciente, especialista de atención médica.
Informe final	<ul style="list-style-type: none"> Elaborar el informe final. Presentación de resultados en sede central. 	Reporte final. Artículo técnico.	Computadora, papel, impresora.

7. RESULTADOS

Como parte del proceso de la investigación, se realiza la presentación y demostración de la funcionalidad y uso del prototipo al especialista en el centro hospitalario, con el objetivo de validar, verificar y comparar las lecturas que realiza el prototipo con las de un equipo o monitor biomédico especializado, el cual se encuentra en uso para tales fines (medición de signos vitales).

En la tabla 4, se presentan los resultados obtenidos de dicho proceso, observando que para las variables de temperatura corporal y oximetría (SpO2 y frecuencia cardiaca) las variaciones o diferencias son insignificantes.

Por otro lado, es evidente que los datos comparativos para la variable de la tensión arterial tienen un margen de error alto. Las pruebas realizadas fueron con un paciente en estado de salud normal.

Tabla 4. Lecturas comparativas del prototipo vs monitor o equipo biomédico especializado en el centro hospitalario.

VARIABLE	LECTURA CON PROTOTIPO	LECTURA CON MONITOR ESPECIALIZADO (M9000)
Temperatura corporal	33.84 °C	33.8 °C
Frecuencia Cardíaca	59 PRbpm	59 PRbpm
Saturación Parcial del Oxígeno	98 %	98 %
Tensión Arterial		
Diastólica/o	104 mmHg	71 mmHg
Sistólica/o	124 mmHg	117 mmHg
Ritmo cardíaco	82 bpm	82 bpm

Diagrama de bloques del sistema

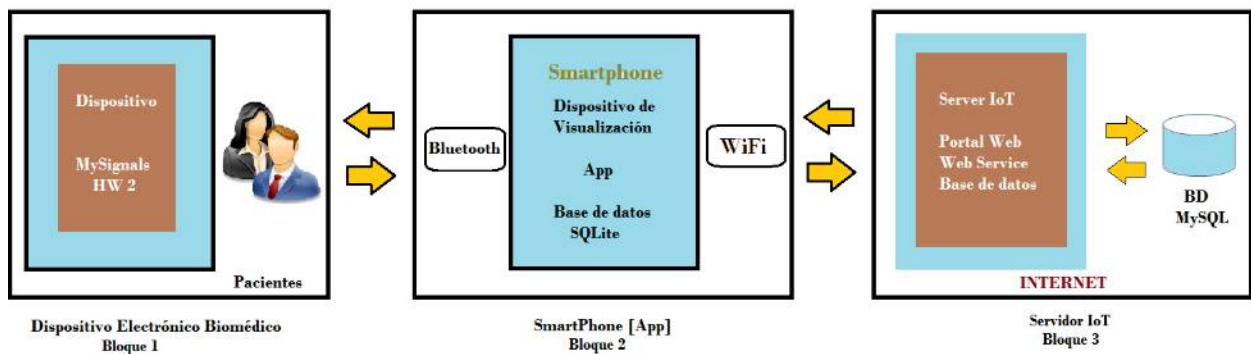


Fig. 4. Diagrama de bloque.

En la Fig. 4, se presenta el diagrama de bloques que describe por un lado las partes que componen el sistema y, por otro lado, las tecnologías utilizadas para el desarrollo del prototipo a nivel de Hardware y Software.

Bloque 1, Dispositivo Electrónico Biomédico. Este bloque está conformado por el dispositivo electrónico, en el cual se conectarán los sensores biométricos y el/los pacientes de los cuales se desean tomar muestras para las variables biométricas antes mencionadas. Cuenta con comunicación vía Bluetooth, USB y UART.

Bloque 2, Smartphone. Este bloque está conformado por el dispositivo inteligente en el cual se instalará la aplicación o App desarrollada (Celular o Tablet), la cual tiene como función:

- Servir como monitor para visualizar las lecturas biométricas realizadas vía Bluetooth.
- Es el responsable de enviar la data mostrada en el monitor (pantalla del Smartphone) al servidor IoT para su procesamiento y almacenamiento en la base de datos en la nube. Función configurable.

- Es el que permite la interacción y comunicación con el dispositivo electrónico y el servidor de IoT en la nube para el almacenamiento remoto de la información clínica de pacientes conectados al dispositivo por medio de la App.
- Administrar las cuentas de usuarios de los especialistas y el/los pacientes a su responsabilidad en base de datos nativa (SQLite).
- Administrar datos personales de pacientes en base de datos nativa (SQLite).

Bloque 3, Servidor IoT. Es el ordenador donde se almacenan los ficheros del portal web, web service y base de datos.

- Portal Web: este permite que cualquier dispositivo inteligente por medio de un navegador web pueda consultar la información clínica de pacientes. Esta es visualización en Dashboard para su fácil visualización e interpretación. Escrito en lenguaje programación PHP, JSON, JAVASCRIPT Y AJAX.
- Web Service: permite la comunicación bidireccional con la base de datos en el servidor IoT con la aplicación móvil (App). Escrito en lenguaje de programación PHP, JSON, JAVASCRIPT.
- Base de datos: contenedor o espacio para almacenamiento de los datos clínicos de los pacientes. Gestos de base de datos utilizado MySQL.

Esquema de conexión de sensores en prototipo final.

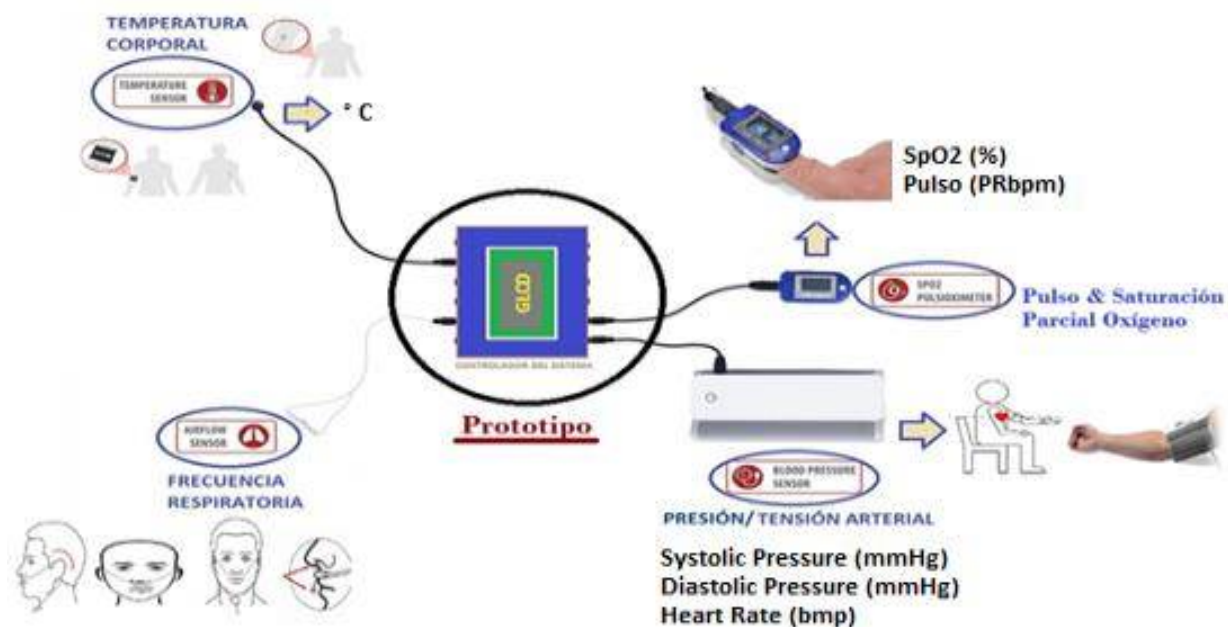


Fig. 5. Prototipo y conexión de sensores biométricos.

En la Fig. 5, se presenta el diseño físico final del prototipo, así como la forma de conexión de cada uno de los sensores biométricos de las variables fisiológicas consideradas y descritas anteriormente para el dispositivo, además se puede apreciar la forma de uso de cada elemento (Ubicación del sensor en el cuerpo, posición o postura del cuerpo para medir una determinada variable y las unidades de medición por variable). Para mayor detalle. Ver apartado Anexos, sección sensores biométricos del sistema.

Diseño electrónico del sistema.

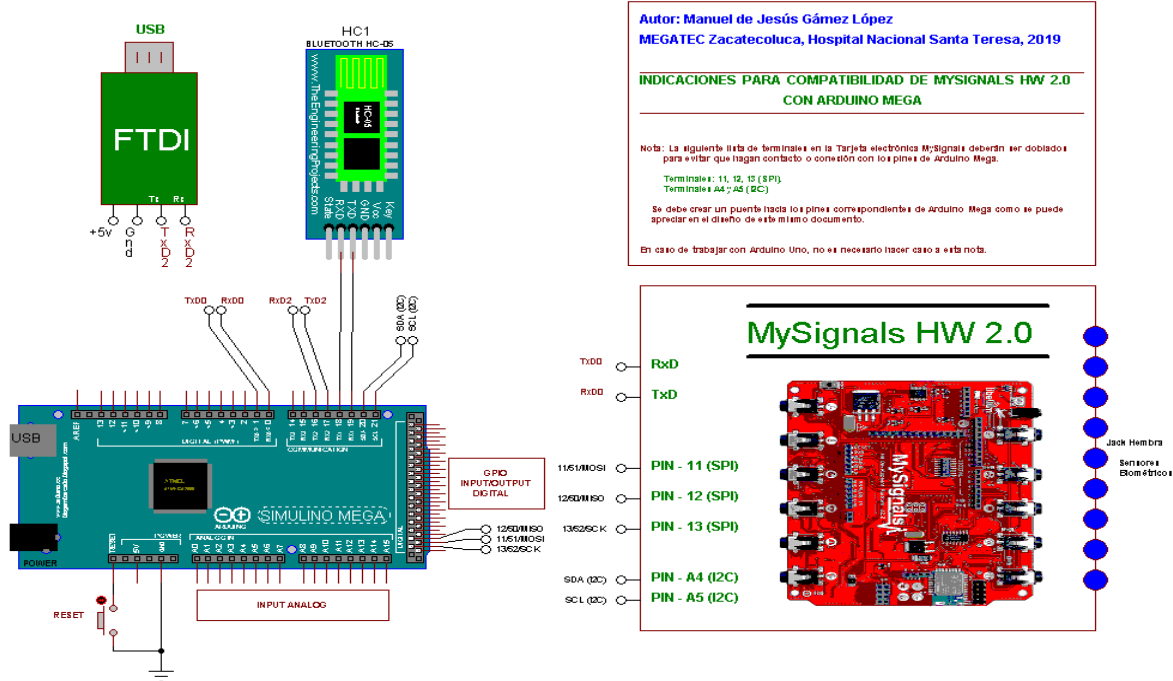


Fig. 6. Diseño electrónico.

La tarjeta electrónica MySignals HW, es compatible directamente con la tarjeta Arduino Uno, la cual tiene muy poca capacidad de memoria de programa y de datos, esto obliga a estar reprogramando el Arduino Uno, por cada prueba de un sensor diferente. Para evitar este hecho y poder utilizar un controlador más potente (Arduino MEGA), o de mayores prestaciones; el cual permita almacenar todo el código fuente necesario para probar cualquier sensor, es necesario realizar todas las modificaciones de diseño electrónico que se pueden apreciar en la Fig. 6. Esto es debido a que el microcontrolador que poseen ambas placas es diferente y por ende cambia la distribución de los pines en cada tarjeta, más, sin embargo, comparten características comunes de protocolos de comunicación y que para este sistema interesaba que tuvieran principalmente tres protocolos que se han utilizado para tal caso: UART (Serie/RS232), I2C y SPI. En la biblioteca de MySignals se puede comprobar uno de los principales cambios que hay que realizar para el bus SPI del sistema. Ver Fig. 7. Por otro lado, para saber los cambios del bus I2C, fue necesario ver el diagrama de terminales de ambos controladores (UNO y MEGA).

```

55 #define MYSIGNALS_DEBUG 0
56
57 // Pin definition
58 #if defined(__AVR_ATmega328P__)
59 #define SPI_MISO 12
60 #define SPI_MOSI 11
61 #define SPI_CLOCK 13
62 #endif
63
64
65
66 #if defined(__AVR_ATmega2560__)
67 #define SPI_MISO 50
68 #define SPI_MOSI 51
69 #define SPI_CLOCK 52
70 #endif
    
```

Fig. 7. Terminales SPI, controlador ATmega328P-PU y ATmega2560.

Para este sistema se utilizó la comunicación serial (Rx/D y Tx/D), líneas de recepción y transmisión respectivamente.

- UART/0 (Tx/D0 / Tx/D0): Este canal es el que permite la comunicación bidireccional directa con la tarjeta MySignals y el controlador del sistema para el intercambio de información de las variables biométricas.
- UART/1 (Tx/D1 / Tx/D1): Este canal es el que permite que el sistema electrónico se comunique vía Bluetooth con la App (Aplicación Móvil para Smartphone y Tablet). Se requiere que ambos dispositivos cuenten con: Bluetooth, WiFi y red GSM (Con chip para notificaciones SMS).
- UART/2 (Tx/D2 / Tx/D2): Este canal es el que permite comunicar por Interface FTDI/USB al sistema con un ordenador. Para ello se debe utilizar programas de virtualización de puertos serie. Por ejemplo: Terminal, teraterm, putty, Coolterm, entre otros.

Nota: Para la pantalla (GLCD) utilizada no se ponen conexiones en el diseño debido a que esta va montada en el slot que ya incluye la misma tarjeta MySignals, y por tanto no fue necesario investigar sobre las conexiones, solo como programarla para mostrar la información. En el diseño se ponen todos los cambios necesarios de conexión para que MySignals trabaje con el controlador Mega, esto por la limitante de capacidad en el controlador que recomienda el fabricante para pruebas.

Diseño de la base de datos.

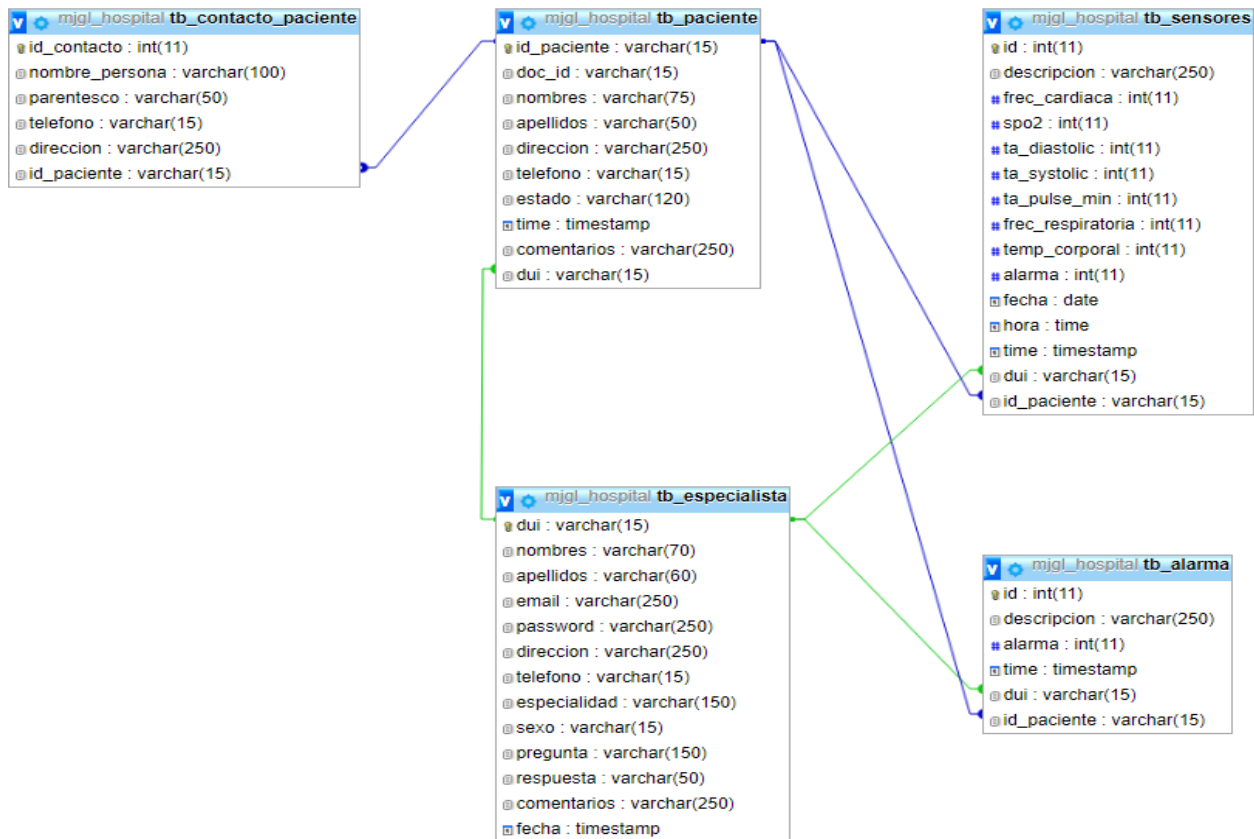


Fig. 8. Base de datos.

Diseño del Portal Web.

A continuación, se presentan los diseños realizados para el portal web que acompaña a esta investigación. La función principal de esta plataforma es permitir el acceso a la información clínica del paciente desde cualquier ubicación por medio de un dispositivo inteligente (PC, Smartphone o Tablet).



Fig. 9. LOGIN.

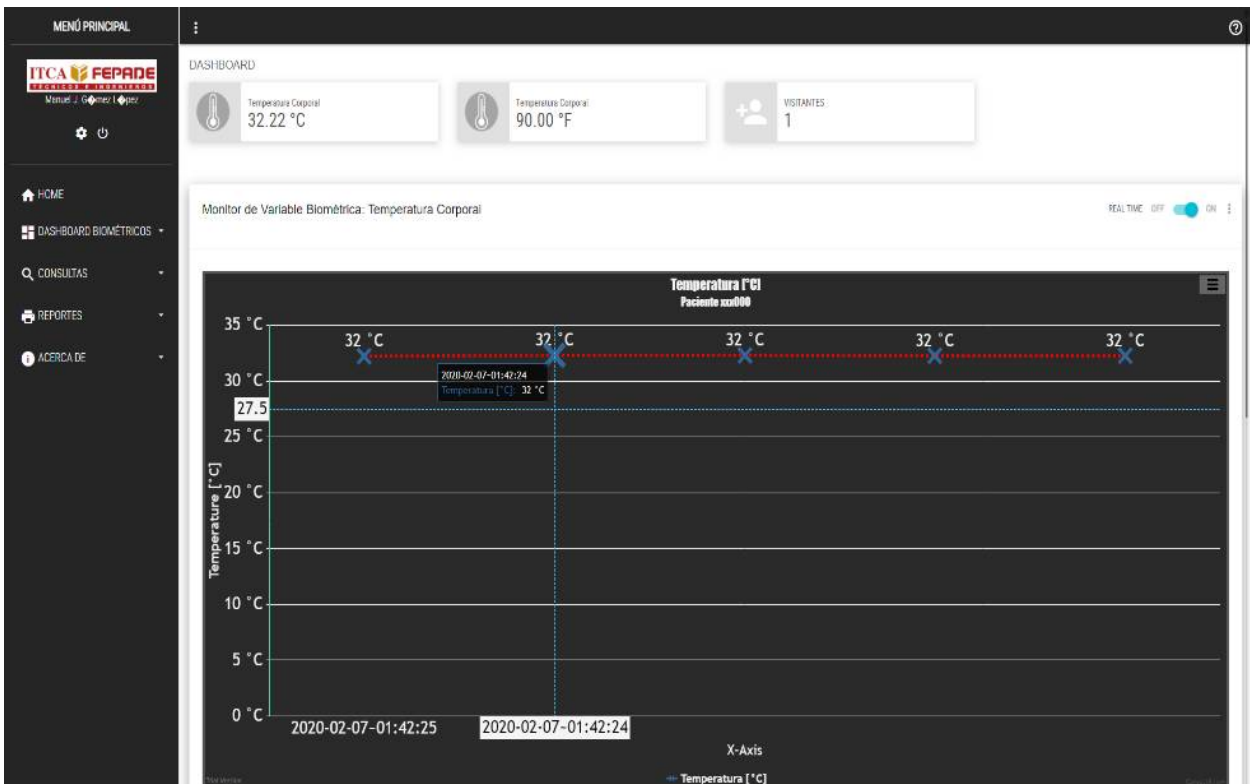


Fig. 10. Dashboard Temperatura Corporal.

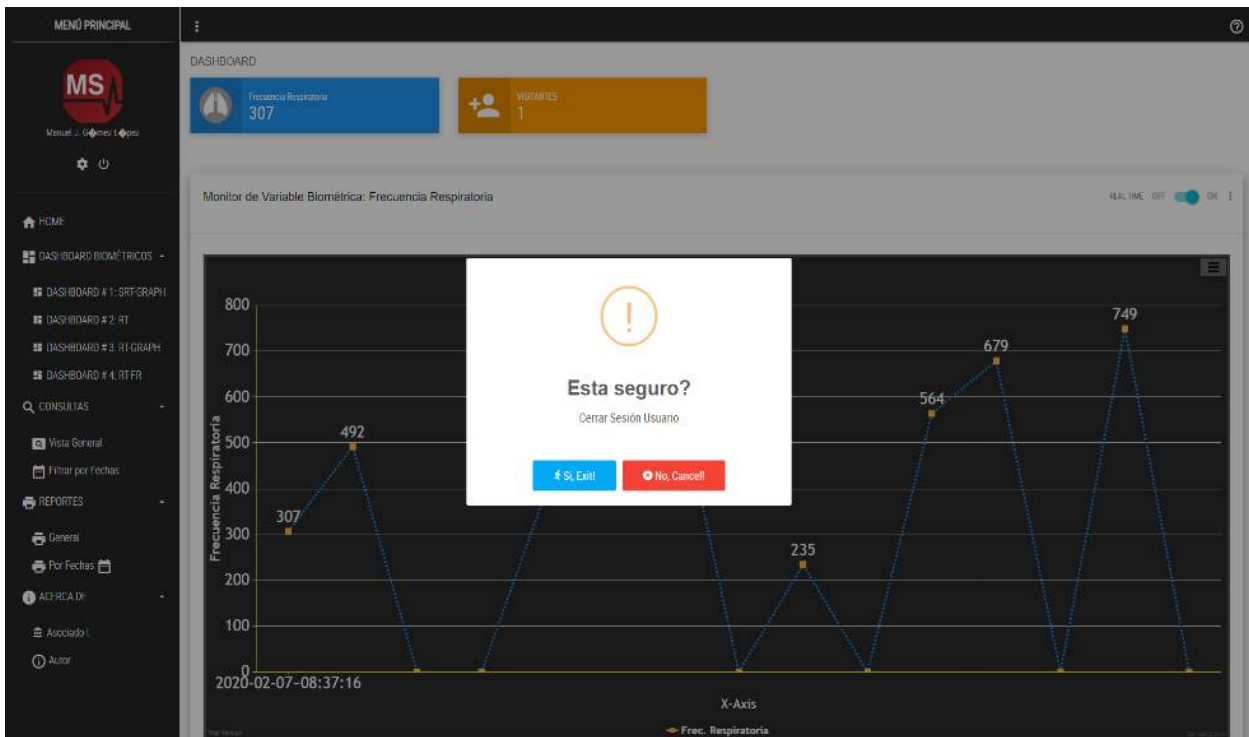


Fig. 11. Frecuencia respiratoria.

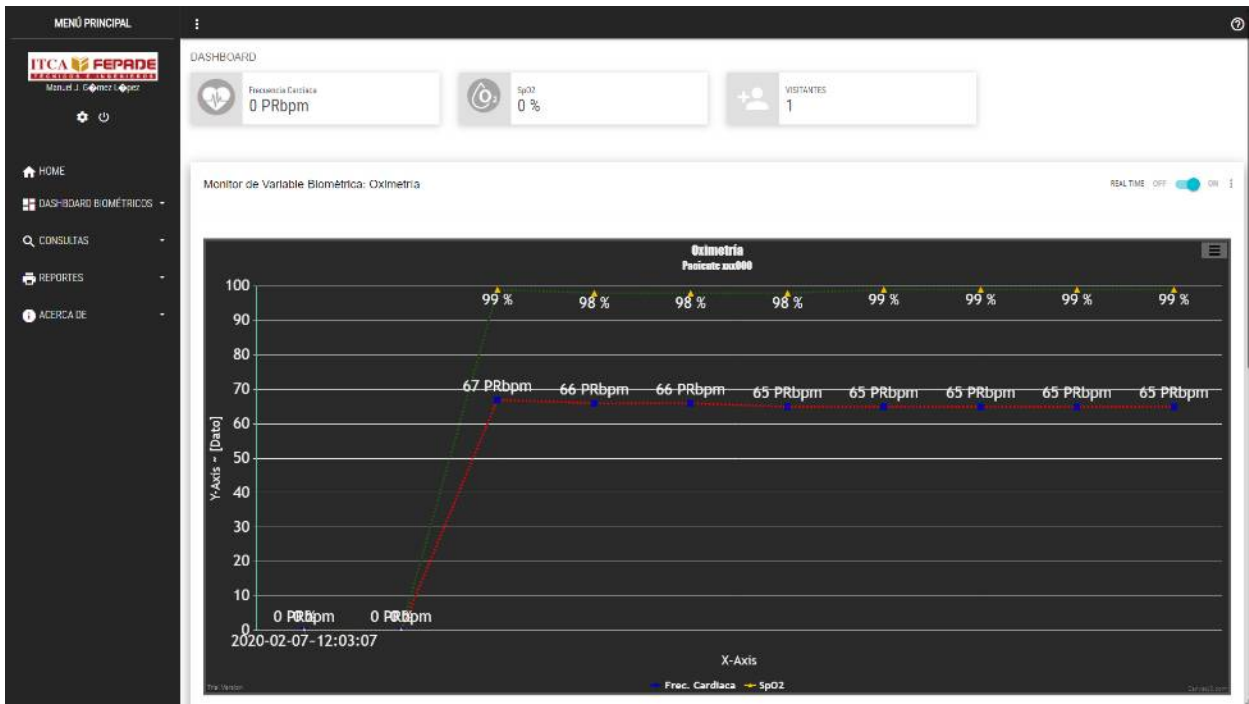


Fig. 12. Dashboard Oximetría.

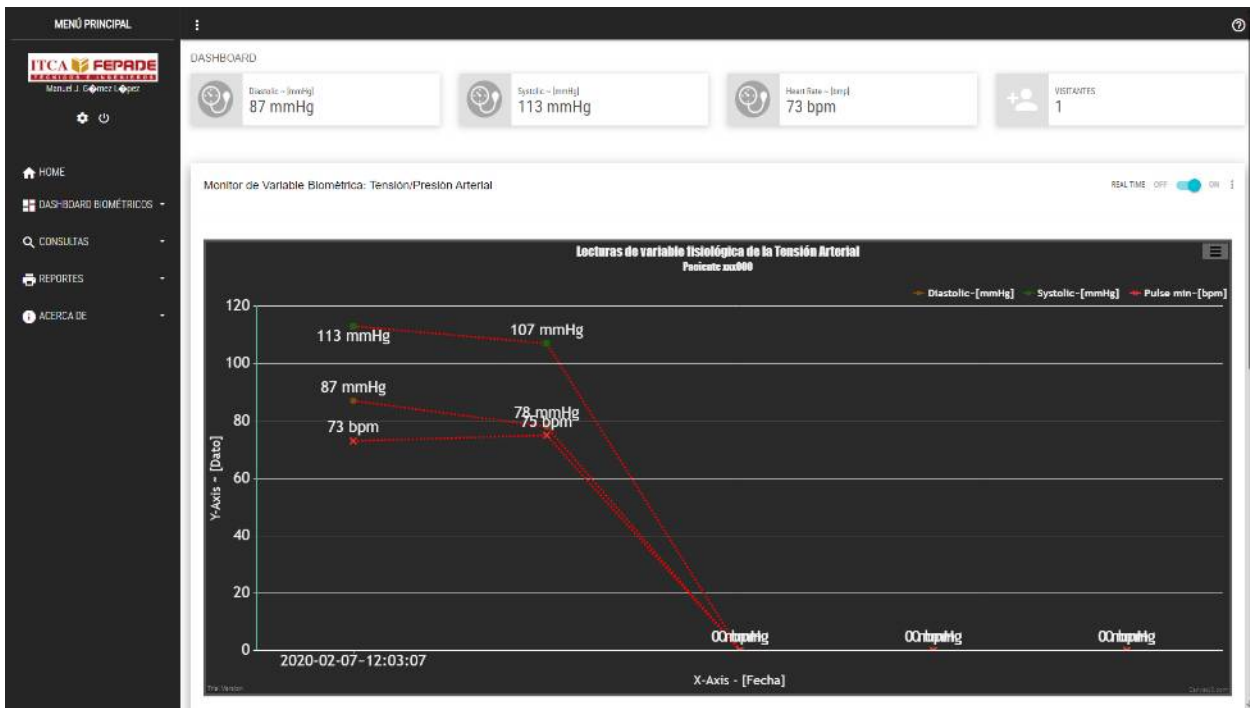


Fig. 13. Dashboard Presión Arterial.



Fig. 14. Dashboard Presión Arterial.

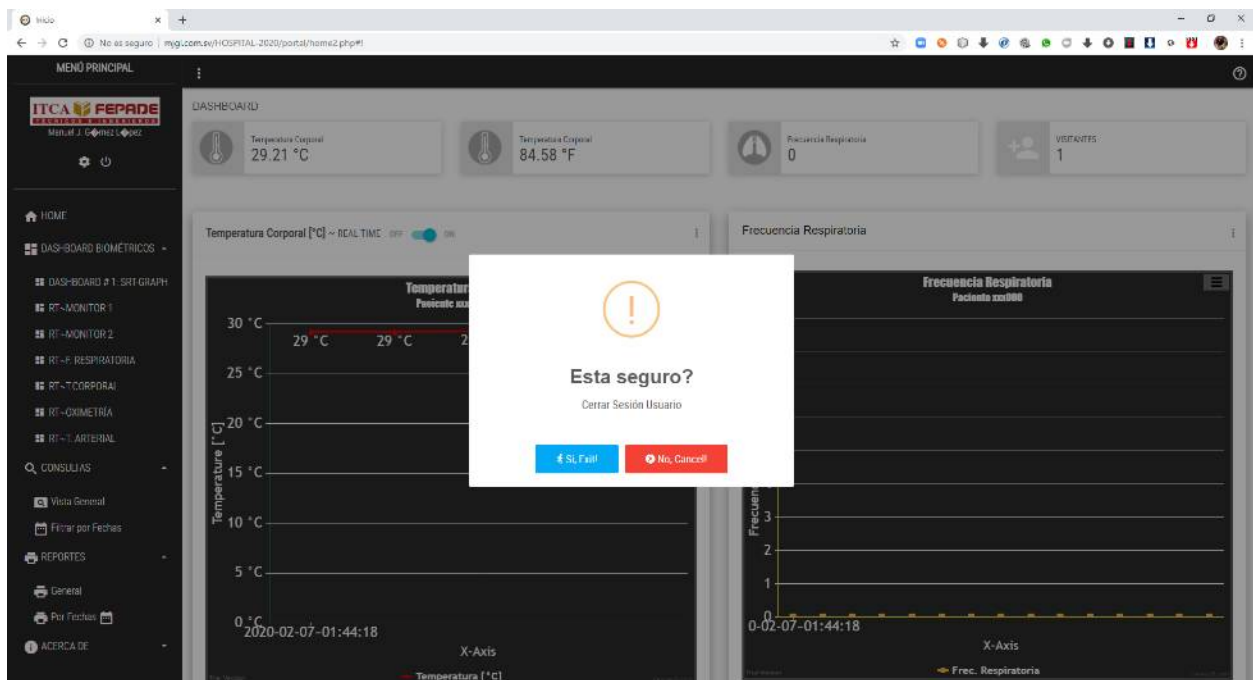


Fig. 15. Dashboard General 1.

MS
Manuel J. Gómez López

Filtrar Rango de Fechas → Historico de Información en la Base de Datos.

Desde: dd/mm/aaaa Hasta: dd/mm/aaaa

N°	Frec. Cardíaca	SpO2	Diastolic	Systolic	Pulse/min	Frec_respiratoria	Temp. Corporal	Alarma	Fecha - Hora
18418	75	98	0	0	0	0	0.00 °C ~ 32.00 °F	1	2020-02-03 03:36:12
18417	76	98	0	0	0	0	0.00 °C ~ 32.00 °F	1	2020-02-03 03:36:02
18416	77	96	0	0	0	0	0.00 °C ~ 32.00 °F	1	2020-02-03 03:35:52
18415	77	96	0	0	0	0	0.00 °C ~ 32.00 °F	1	2020-02-03 03:35:42
18414	75	96	0	0	0	0	0.00 °C ~ 32.00 °F	1	2020-02-03 03:35:32
18413	77	99	0	0	0	0	0.00 °C ~ 32.00 °F	1	2020-02-03 03:35:22
18412	79	98	0	0	0	0	0.00 °C ~ 32.00 °F	1	2020-02-03 03:35:12
18411	78	98	0	0	0	0	0.00 °C ~ 32.00 °F	1	2020-02-03 03:35:02
18410	73	99	0	0	0	0	0.00 °C ~ 32.00 °F	1	2020-02-03 03:34:52
18409	75	99	0	0	0	0	0.00 °C ~ 32.00 °F	1	2020-02-03 03:34:42

PRbpm %SpO2 mmHg mmHg bpm Status [°C ~ °F] Status Time

Fig. 16. Búsqueda filtrada por fecha.

Consulta Lecturas de Sensores

Show 10 entries

N°	Frec. Cardiaca	SpO2	Diastolic	Systolic	Pulse/min	Frec. respiratoria	Temp. Corporal	Alarma	Fecha - Hora
12651	0	0	71	107	80	0	0.00	1	2020-01-24 02:45:40
12652	0	0	67	98	74	0	0.00	1	2020-01-24 02:46:38
12653	0	0	61	107	69	0	0.00	1	2020-01-24 02:47:25
12654	0	0	0	0	0	0	0.00	0	2020-01-27 01:40:16
12655	0	0	0	0	0	0	0.00	0	2020-01-27 01:40:17
12656	0	0	0	0	0	0	0.00	0	2020-01-27 01:40:18
12657	0	0	0	0	0	0	0.00	0	2020-01-27 01:40:19
12658	0	0	0	0	0	0	0.00	0	2020-01-27 01:40:20
12659	0	0	0	0	0	0	0.00	0	2020-01-27 01:40:21
12660	0	0	0	0	0	0	0.00	0	2020-01-27 01:40:24

Showing 1 to 10 of 5,768 entries

Fig. 17. Consulta general.

Filtrar Reporte por Fechas

Seleccione Rango de Fechas

Desde: Hasta:

TEMA DE LA INVESTIGACIÓN: [Visita Nuestras Redes Sociales](#) [FACEBOOK](#)

Fig. 18. Reporte filtrado por fecha.

Reporte general

Showing 1 to 10 of 5,768 entries

N°	Frec. Cardiaca	SpO2	Diastolic	Systolic	Pulse/min	Frec. respiratoria	Temp. Corporal	Alarma	Fecha - Hora
12651	0	0	71	107	80	0	0.00 °C - 32.00 °F	1	2020-01-24 02:45:40
12652	0	0	67	98	74	0	0.00 °C - 32.00 °F	1	2020-01-24 02:46:38
12653	0	0	61	107	69	0	0.00 °C - 32.00 °F	1	2020-01-24 02:47:25
12654	0	0	0	0	0	0	0.00 °C - 32.00 °F	0	2020-01-27 01:40:16
12655	0	0	0	0	0	0	0.00 °C - 32.00 °F	0	2020-01-27 01:40:17
12656	0	0	0	0	0	0	0.00 °C - 32.00 °F	0	2020-01-27 01:40:18
12657	0	0	0	0	0	0	0.00 °C - 32.00 °F	0	2020-01-27 01:40:19
12658	0	0	0	0	0	0	0.00 °C - 32.00 °F	0	2020-01-27 01:40:20
12659	0	0	0	0	0	0	0.00 °C - 32.00 °F	0	2020-01-27 01:40:21
12660	0	0	0	0	0	0	0.00 °C - 32.00 °F	0	2020-01-27 01:40:24

Fig. 19. Reporte general.

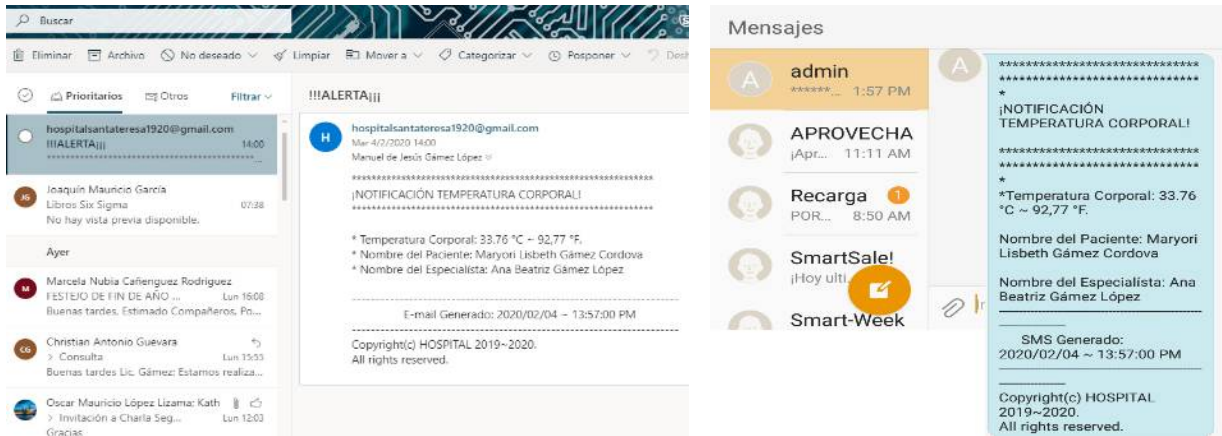


Fig. 20. Notificaciones E-mail y SMS.

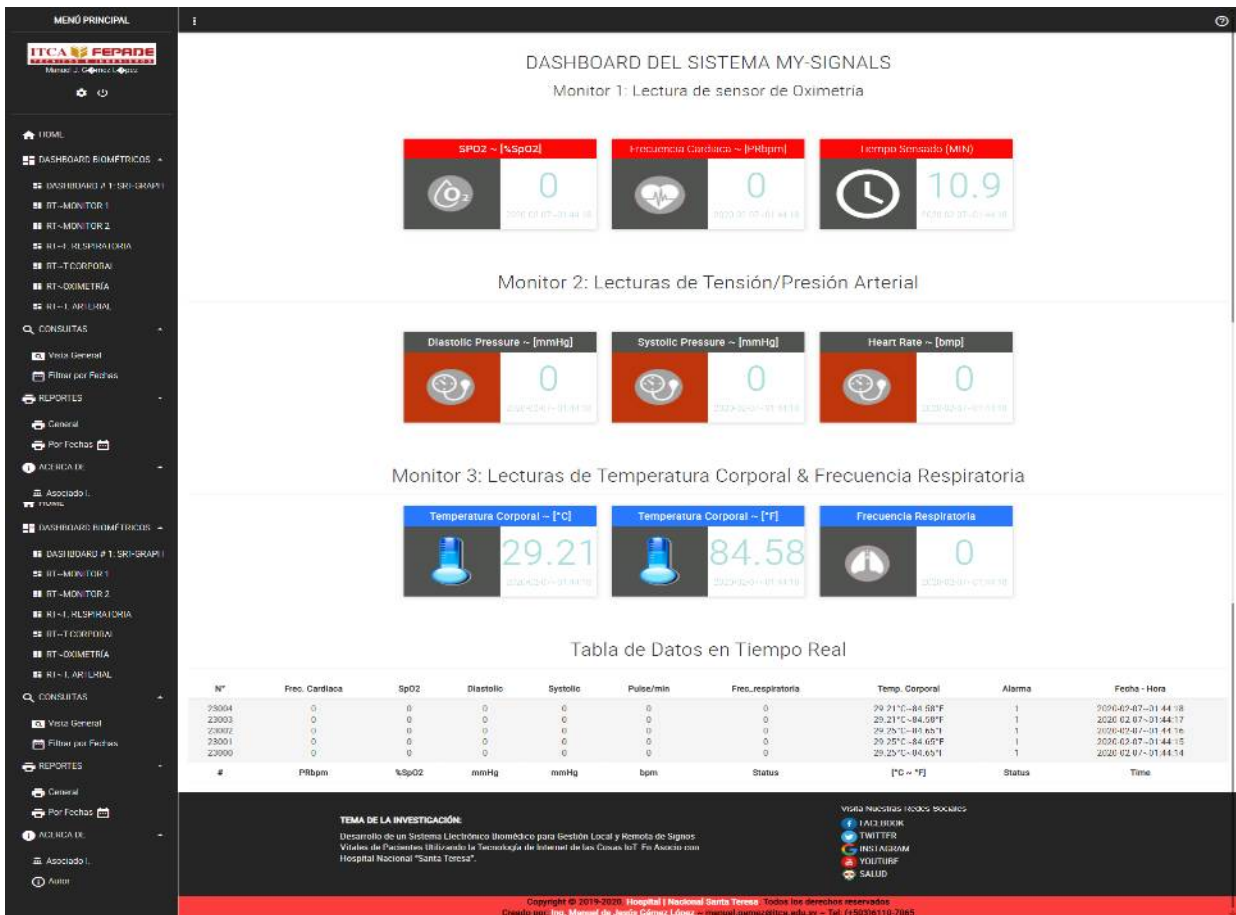


Fig. 21. Dashboard general 2.

Diseño de la Aplicación Móvil (App para Smartphone y Tablet).

A continuación, se presentan las ventanas principales que se diseñaron para la App. Ver Fig. 22 a Fig. 28.



Fig. 22. LOGIN y registros.

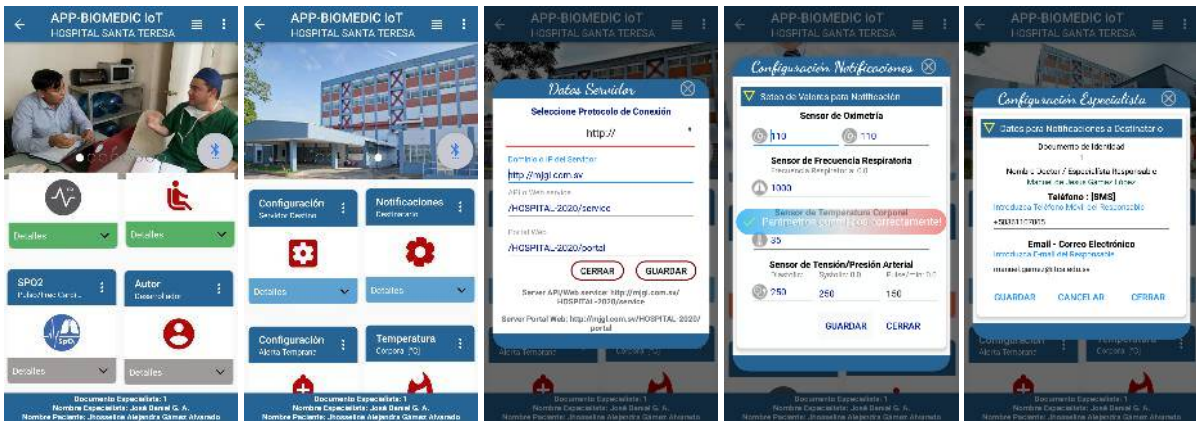


Fig. 23. Menú principal y ventanas de configuración.

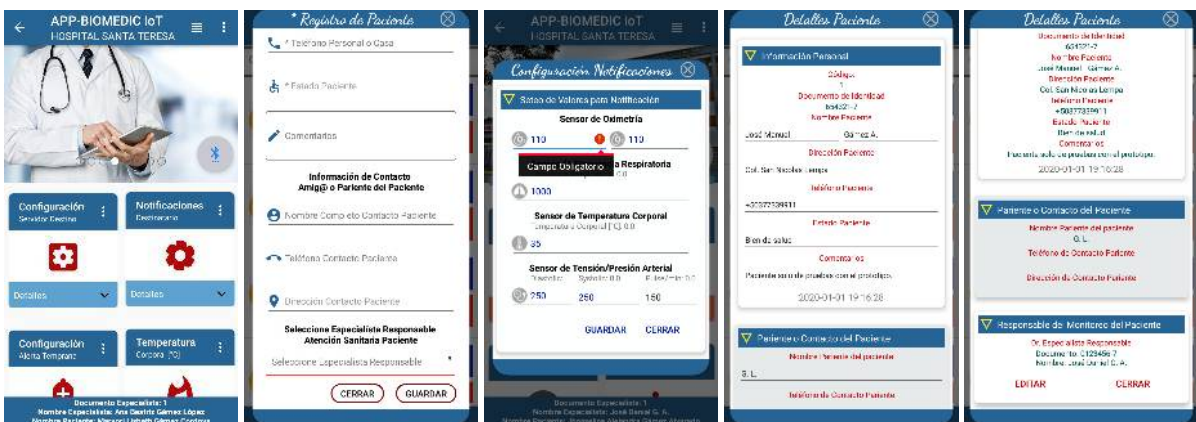


Fig. 24. Registros y edición de información.

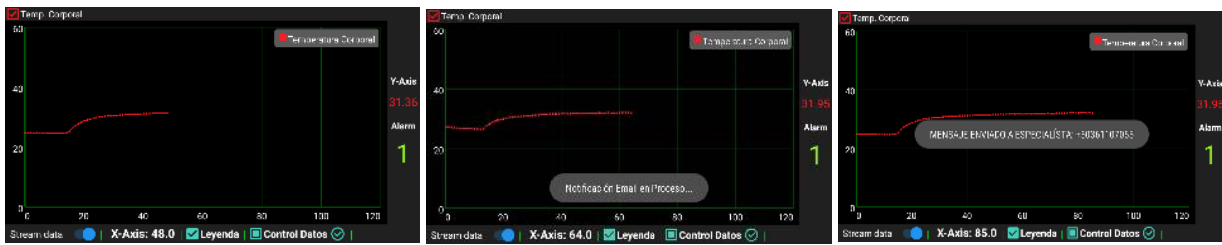


Fig. 25. Monitor temperatura corporal.

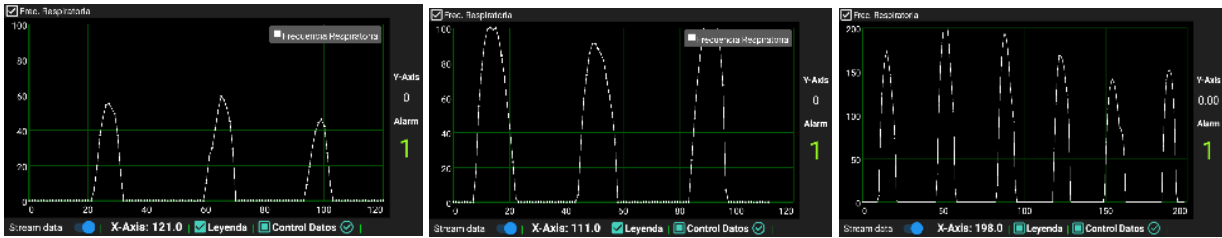


Fig. 26. Monitor frecuencia respiratoria.

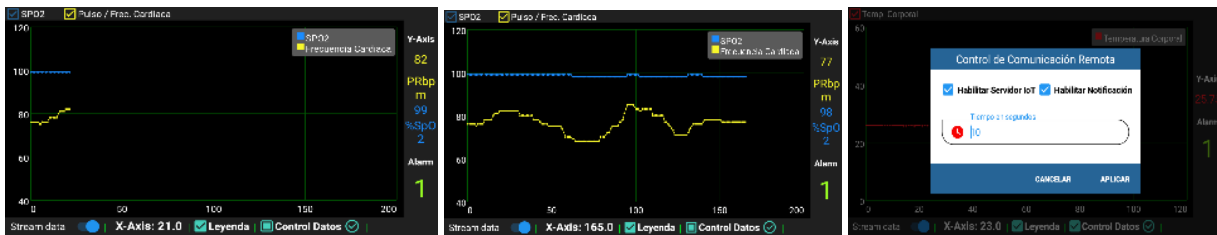


Fig. 27. Monitor oximetría (SpO2 y pulso).

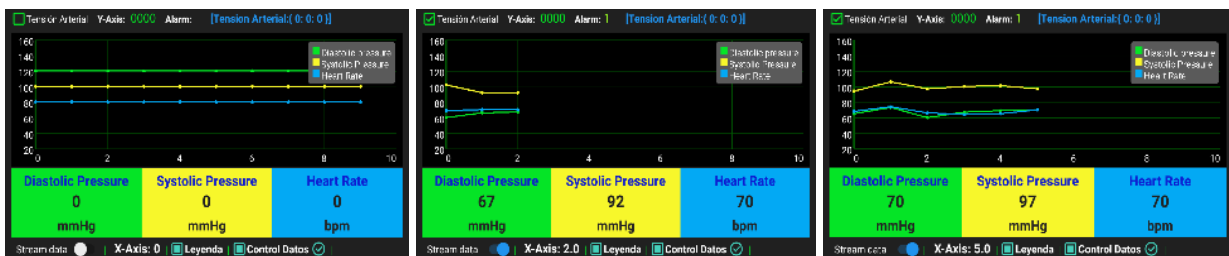


Fig. 28. Monitor presión arterial.

8. CONCLUSIONES

- El sistema en general presenta un retardo, tiempo de estabilización para su correcto funcionamiento y lecturas de los sensores, se observa en la información de resultados que luego de pasar unos segundos, el equipo toma valores muy cercanos a los de un equipo especializado, esto se debe a que el dispositivo necesita de una estabilización para proporcionar lecturas más precisas.
- Una vez realizadas pruebas con personal del Hospital Santa Teresa de Zacatecoluca, se pudo establecer que el prototipo arroja valores muy cercanos en comparación con los de un equipo especializado. Proporcionando este un grado de confiabilidad del 95% para las lecturas de

temperatura corporal, saturación parcial del oxígeno y frecuencia cardiaca. Para la variable frecuencia respiratoria no se pudo efectuar comparaciones debido a que dicha medición la toman basada en otro método a través de la conexión de electrodos, siendo un sensor diferente al de esta investigación. En la medición realizada para la comparación de la variable biométrica tensión arterial, sí se detectaron variaciones o diferencias en las lecturas.

- El prototipo desarrollado aplica para usos de personas en estado no crítico y mayores de edad (mayores 18). Puede ser utilizado para casos de uso domésticos, contando con la ventaja tecnológica de poder enviar la data de las mediciones biométricas a un servidor (opción configurable desde App), para su acceso desde cualquier ubicación del mundo a través de dispositivos inteligentes.
- El potencial que representan los dispositivos móviles en el desarrollo de proyectos para el área de investigación y específicamente por las tecnologías de comunicación y de desarrollo que este incorpora de fábrica son exageradamente enormes, volviéndose la programación el centro crítico necesario u obligatorio para el uso y explotación de estas tecnologías.

9. RECOMENDACIONES

- Para el uso del prototipo y el Smartphone es recomendable contar con un sistema de backup de energía (UPS) por apagones, debido a que, si no se cuenta con ello, por apagón en caso de estar en una ventana de monitor se perderá la comunicación con los diferentes componentes del sistema.
- Alimentar el equipo o dispositivo electrónico con la fuente de tensión suministrada con este, para disponer de un correcto funcionamiento del mismo.
- Para el uso de prototipo es de vital importancia, comprobar y asegurarse que el dispositivo móvil (Smartphone o Tablet) permanezca conectado a una red WiFi segura con conexión a Internet o tener un plan de datos activo, esto para que el sistema envíe sin inconvenientes la información al servidor de destino. En caso de ser el servidor un ordenador dentro de una intranet o local no es necesario que la red WiFi cuente con conectividad a Internet, ni plan de datos activado.
- Para el proceso de medición de la tensión arterial, es importante que la colocación del sensor en el brazo y el cuerpo se mantengan en la posición correcta y sin realizar movimientos bruscos, esto permite que se puedan obtener mediciones más confiables.
- Buscar la mejora de los procedimientos en los servicios brindados en atención sanitaria, incorporando la tecnología en los equipamientos biomédicos especializados con los que se cuenta en los centros hospitalarios públicos y privados, para convertir el rutinario procedimiento desfasado en espacios sofisticados y tecnológicos de gestión y atención a pacientes.
- Trabajar el sistema e-health de MySignals HW con el controlador Mega u otro más avanzado en procesamiento, capacidad de memoria para programa y de datos. Esto permite al sistema hacer mediciones múltiples sin tener que estar reprogramando el controlador por medición, además, esto permite la incorporación de otros escudos o dispositivos de comunicación y periféricos como, por ejemplo, una pantalla TFT para visualización de información procesada por el sistema.

10. GLOSARIO

- **Equipo biomédico.** Cualquier instrumento, aparato, implemento, máquina, implante, software, calibrador, marial y otro artículo similar o relacionado destinado por el fabricante para ser utilizado solo o en combinación, en seres humanos, para uno o más de los propósitos específicos de:
 - Diagnóstico, prevención, control y tratamiento o alivio de una enfermedad.
 - Diagnóstico, control, tratamiento y alivio o compensación de una lesión.
 - Investigación, reemplazo, modificación y soporte de la anatomía o de un proceso fisiológico.
 - Apoyo o preservación de la vida, etc.
- **UART.** Son las siglas de Universal Asynchronous Receiver-Transmitter o Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo.
- **SPI.** El Bus SPI (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj (comunicación sincrónica).
- **I2C.** Circuito inter-integrado (I²C, del inglés Inter-Integrated Circuit) es un bus serie de datos desarrollado en 1982 por Philips Semiconductors. El sistema original fue desarrollado por Philips a principios de 1980 con el fin de controlar varios chips en televisores de manera sencilla.
- **E-Health o e-Salud.** Es el término con el que se define al conjunto de Tecnologías de la Información y la Comunicación (TICs) que, a modo de herramientas, se emplean en el entorno sanitario en materia de prevención, diagnóstico, tratamiento, seguimiento, así como en la gestión de la salud, ahorrando costes al sistema sanitario y mejorando la eficacia de este.
- **Smartphone o Teléfono celular.** Dispositivo inteligente que permite al usuario conectarse a Internet, gestionar cuentas de correo electrónico e instalar otras aplicaciones y recursos a modo de pequeño computador.
- **Atención médica.** Se conoce como asistencia sanitaria en España y como atención médica, asistencia médica, atención sanitaria, o atención de salud en algunos países latinoamericanos, al conjunto de servicios que se proporcionan al individuo, con el fin de promover, proteger y restaurar su salud.
- **Patología.** Rama de la medicina que se enfoca en las enfermedades del ser humano y, el otro, cómo el grupo de síntomas asociadas a una determinada dolencia.
- **m-Health.** Término que se refiere al uso de dispositivos móviles para mejorar la atención de los pacientes.
- **Telemática.** Es la disciplina científica y tecnológica que analiza e implementa servicios y aplicaciones que usan tanto los sistemas informáticos como los sistemas de telecomunicación, como resultado de la unión de ambas disciplinas. Por ejemplo: Cualquier tipo de comunicación a través de Internet o los sistemas de posicionamiento global.

- **Fisiología.** Es el estudio científico de las funciones o mecanismos que funcionan dentro de un sistema vivo. Es uno de los cimientos sobre los cuales se han construido todas las ciencias biológicas y médicas.
- **Telemedicina (distancia + medicina).** Es la prestación de servicios médicos a distancia. Para su implantación se emplean tecnologías de la información y las comunicaciones.
- **Computación ubicua.** Es un concepto en ingeniería de software y las ciencias de la computación. Es entendida como la integración de la informática en el entorno de la persona, de forma que los ordenadores no se perciban como objetos diferenciados, apareciendo en cualquier lugar y en cualquier momento.
- **Biometría.** Es la ciencia y la tecnología dedicada a medir y analizar datos biológicos. En el terreno de la tecnología de la información, la biometría hace referencia a las tecnologías que miden y analizan las características del cuerpo humano, como el ADN, las huellas dactilares, la retina y el iris de los ojos, los patrones faciales o de la voz y las medidas de las manos a efectos de autenticación de identidades. Ejemplo de dispositivos biométricos son los escáneres de huellas dactilares.

Es una tecnología basada en la identificación de las personas a través de algoritmos de características personales, físicas y morfológicas que son únicas y distintivas de cada individuo.

- **Biométrica.** Ciencia enfocada en la identificación de las personas, que ha servido como base para la creación de los marcadores biométricos.
- **Lector o marcador biométrico.** Es un Hardware que permite la identificación de una persona mediante rasgos del cuerpo humano como huellas digitales, iris de los ojos, la voz humana o el rostro contra una base de datos de personas.
- **Señal bioeléctrica.** Es la reacción electroquímica que produce cierto tipo de células al ser excitadas.
- **Hipoxemia.** Es una disminución anormal de la presión parcial de oxígeno en la sangre arterial por debajo de 60 mmHg. También se puede definir como una saturación de oxígeno menor de 90,7%. No debe confundirse con hipoxia, una disminución de la difusión de oxígeno en los tejidos y en la célula.
- **Apnea.** La apnea obstructiva del sueño es una patología respiratoria que se caracteriza porque las personas que lo padecen roncan habitualmente, tienen paradas de la respiración (apneas) de forma repetida y somnolencia durante el día.

11. REFERENCIAS BIBLIOGRÁFICAS

- [1]. C. Hacks, "e-Health Sensor Platform V2. 0 for Arduino and Raspberry Pi [Biometric/Medical Applications]," Recuperado el, vol. 24, 2014.
- [2]. Libelium, (2017, febrero 10). Plataforma de desarrollo de IoT médica y eHealth para Arduino (MySignals HW v2). [Online]. Available: <https://www.cooking-hacks.com/mysignals-hw-ehealth-medical-biometric-iot-platform-arduino-tutorial/>
- [3]. Libelium, (2017, febrero 10). MySignals HW Complete Kit (plataforma de desarrollo médico de eHealth para Arduino) (MySignals HW v2). [Online]. Available: <https://www.cooking-hacks.com/mysignals-hw-ehealth-medical-biometric-arduino-complete-kit/>

12. ANEXOS

Sensores biométricos del sistema.

1. Pulso y oxígeno en sangre (SPO2)

La oximetría de pulso es un método no invasivo para indicar la saturación de oxígeno arterial de la hemoglobina funcional.

La saturación de oxígeno se define como la medición de la cantidad de oxígeno disuelto en la sangre, basada en la detección de hemoglobina y desoxihemoglobina. Se utilizan dos longitudes de onda de luz diferentes para medir la diferencia real en los espectros de absorción de HbO₂ y Hb. El torrente sanguíneo se ve afectado por la concentración de HbO₂ y Hb, y sus coeficientes de absorción se miden utilizando dos longitudes de onda de 660 nm (espectros de luz roja) y 940 nm (espectros de luz infrarroja). La hemoglobina desoxigenada y oxigenada absorbe diferentes longitudes de onda.



Fig. 29. Sensor oximetría.

La hemoglobina desoxigenada (Hb) tiene una absorción más alta a 660 nm y la hemoglobina oxigenada (HbO₂) tiene una absorción más alta a 940 nm. Luego, un fotodetector percibe la luz no absorbida de los LED para calcular la saturación de oxígeno arterial.

Un sensor de oxímetro de pulso es útil en cualquier entorno en el que la oxigenación de un paciente sea inestable, incluidos los cuidados intensivos, la operación, la recuperación, la emergencia y la sala de hospital, los pilotos en aeronaves sin presión, para evaluar la oxigenación de cualquier paciente y determinar la efectividad o la necesidad oxígeno suplementario.

Los rangos normales aceptables para los pacientes son de 95 a 99 por ciento, aquellos con un problema de impulso hipóxico esperarían valores entre 88 y 94 por ciento, valores de 100 por ciento pueden indicar intoxicación por monóxido de carbono.

El sensor debe estar conectado al conector Jack SPO2 específico en la placa MySignals y funciona con una fuente de alimentación de conector directo.

Tabla 5. Rango de medición.

Parámetro	Unidad	Rango
Pulso	ppm	25 ~ 250 ppm
SPO2	%	35-100%

Conexión del Sensor

Conecte el sensor en el conector SPO2 indicado en la placa de Hardware de MySignals. El cable del sensor solo tiene una forma de conexión para evitar errores y facilitar la conexión. Use el conector mini-USB para vincular el SPO2 con la placa MySignals, usando el conector Jack del cable en este lado.



Fig. 30. Conexión Sensor oximetría en placa MySignals.

Coloque el SPO2 en el dedo índice como se muestra en la Fig. 31.



Fig. 31. Conexión y medición de variable biométrica SpO2 y pulso.

Después de unos segundos, obtendrá los valores en la pantalla del sensor, TFT y pantalla de Smartphone.

2. Sensor de Temperatura Corporal.

La temperatura corporal depende del lugar del cuerpo en el que se realiza la medición, y la hora del día y el nivel de actividad de la persona. Diferentes partes del cuerpo tienen diferentes temperaturas.

La temperatura corporal central promedio comúnmente aceptada (tomada internamente) es 37.0 ° C (98.6 ° F). En adultos sanos, la temperatura corporal fluctúa aproximadamente 0.5 ° C (0.9 ° F) durante todo el día, con temperaturas más bajas por la mañana y temperaturas más altas al final de la tarde y noche, a medida que cambian las necesidades y actividades del cuerpo.

El sensor de temperatura le permite medir este parámetro clave para la monitorización del cuerpo. Mide la temperatura del cuerpo humano con una desviación máxima de 0.1 °C.



Fig. 32. Sensor Temperatura Corporal.

Es de gran importancia médica medir la temperatura corporal. La razón es que una serie de enfermedades se acompañan de cambios característicos en la temperatura corporal. Del mismo modo, el curso de ciertas enfermedades se puede controlar midiendo la temperatura corporal, y el médico puede evaluar la eficacia de un tratamiento iniciado.

Hipotermia: <35.0 ° C (95.0 ° F)

Normal: 36.5–37.5 ° C (97.7–99.5 ° F)

Fiebre o hipertermia > 37.5–38.3 ° C (99.5–100.9 ° F)

Hiperpirexia > 40.0–41.5 ° C (104 –106.7 ° F)

El sensor debe estar conectado al Jack o conector de temperatura correspondiente en la placa MySignals y funciona con una fuente de alimentación de conector directo.

Tabla 6. Rango de Medición

Parámetro	Unidad	Distancia
Temperatura corporal	Grado Celsius (° C)	0-50 °C

Conexión del Sensor

Conecte el sensor en el conector de temperatura indicado en la placa de hardware de MySignals. El cable del sensor solo tiene una forma de conexión para evitar errores y facilitar la conexión.

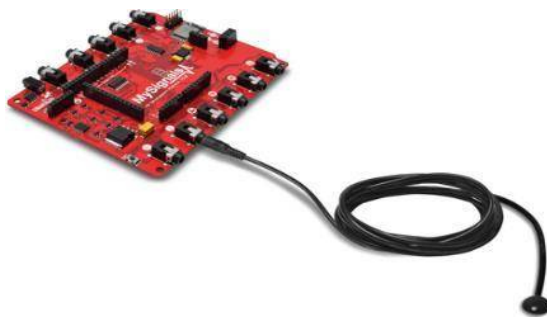


Fig. 33. Conexión del sensor.

Coloque el sensor como se muestra en la Fig. 34.



Fig. 34. Medición de Temperatura Corporal.

Haga contacto entre la parte metálica y su piel. Puede usarse un trozo de cinta adhesiva para sostener el sensor unido a la piel. Ver Fig. 35.



Fig. 35. Medición de temperatura corporal con cinta adhesiva.

Después de unos segundos, obtendrá los valores en la pantalla TFT y pantalla de Smartphone.

NOTA: Si no se miden los datos de un sensor apropiado en el dedo, es necesario usar otras áreas más sensibles del cuerpo como la axila. Ver Fig. 36.

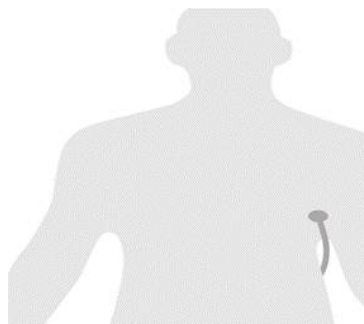


Fig. 36. Medición de temperatura en axila.

3. Sensor de Flujo de Aire.

Las frecuencias respiratorias anormales y los cambios en la frecuencia respiratoria son un indicador amplio de inestabilidad fisiológica importante y, en muchos casos, la frecuencia respiratoria es uno de los primeros indicadores de esta inestabilidad. Por lo tanto, es crítico monitorear la frecuencia respiratoria como un indicador del estado del paciente. El sensor frecuencia respiratoria (AirFlow) puede proporcionar una advertencia temprana de hipoxemia y apnea.

El sensor de flujo de aire nasal / boca es un dispositivo utilizado para medir la frecuencia respiratoria en un paciente que necesita ayuda respiratoria o una persona. Este dispositivo consta de un hilo flexible que se ajusta detrás de las orejas y un conjunto de dos puntas que se colocan en las fosas nasales. La respiración se mide con estos dientes. La cánula / soporte específicamente diseñada permite que el sensor de termopar se coloque en la posición óptima para detectar con precisión los cambios del flujo de aire térmico oral / nasal, así como la temperatura del aire nasal. Cómodo ajustable y fácil de instalar.



Fig. 37. Sensor Frecuencia Respiratoria.

El sensor debe estar conectado al conector de toma de flujo de aire específico en la placa MySignals y funciona con una fuente de alimentación de conector directo.

Tabla 7. Rango de Medición.

Parámetro	Unidad	Distancia
Frecuencia respiratoria	PPM (picos por minuto)	0-60 ppm
Intensidad respiratoria	Voltios	0-3,3V

Conecte el sensor en el Jack de flujo de aire indicado en la placa de Hardware de MySignals. El cable del sensor incluye 2 piezas y solo tiene una forma de conexión para evitar errores y facilitar la conexión.

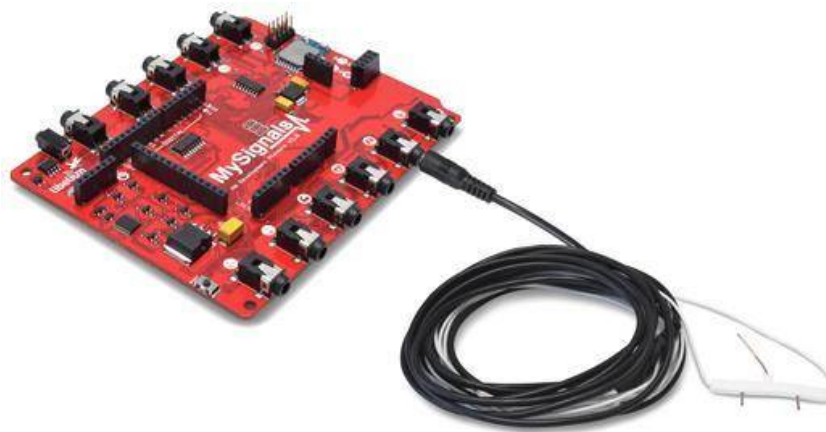


Fig. 38. Conexión del sensor.

El sensor integra un cable de extensión con un conector de "ojo de cerradura". Este conector central tiene una posición específica para tener la polaridad correcta. Verifique las marcas incluidas en el costado de ambos conectores. Ver Fig. 39.



Fig. 39. Conector de "ojo de cerradura".

Coloque el sensor como se muestra en la Fig. 40.

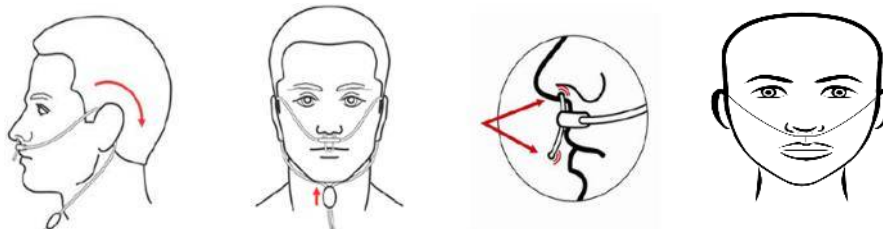


Fig. 40. Posición del sensor para medición.

Después de unos segundos, obtendrá los valores en la pantalla TFT y pantalla de Smartphone.

NOTA: Coloque el sensor en la posición correcta. Ver Fig. 40. Esperar de entre 3-5 minutos para estabilizar la medición del sensor.

4. Sensor de Tensión Arterial.

La presión arterial es la presión de la sangre en las arterias cuando el corazón la bombea alrededor del cuerpo. Cuando su corazón late, se contrae y empuja la sangre a través de las arterias hacia el resto de su cuerpo. Esta fuerza crea presión sobre las arterias. La presión arterial se registra como dos números: la presión sistólica (cuando el corazón late) sobre la presión diastólica (cuando el corazón se relaja entre latidos).

Controlar la presión arterial en el hogar es importante para muchas personas, especialmente si tiene presión arterial alta. La presión arterial no se mantiene igual todo el tiempo. Cambia para satisfacer las necesidades de su cuerpo. Se ve afectado por varios factores, como la posición del cuerpo, la respiración o el estado emocional, el ejercicio y el sueño. Es mejor medir la presión arterial cuando está relajado y sentado o acostado.



Fig. 41. Sensor de Tensión Arterial.

Tabla 8. Clasificación de la presión arterial para adultos (mayores de 18 años).

	Sistólica (mm Hg)	Diastólica (mm Hg)
Hipotensión	<90	<60
Deseado	90-119	60-79
Prehipertensión	120-139	80-89
Etapa 1 Hipertensión	140-159	90-99
Etapa 2 Hipertensión	160-179	100-109
Crisis hipertensiva	≥ 180	≥ 110

La presión arterial alta (hipertensión) puede provocar problemas graves como ataque cardíaco, accidente cerebrovascular o enfermedad renal. La presión arterial alta generalmente no tiene ningún síntoma, por lo que debe controlarse la presión arterial regularmente.

Dentro de los parámetros de Medición del sensor se tienen: presión sistólica, presión diastólica y pulso.

Especificaciones clave

- Método de medición: sistema oscilométrico.
- Rango de medición: presión 0-300 mmHg.
- Pulso 30 ~ 200 p /min.
- Precisión de medición: Presión $\leq \pm 3$ mmHg.
- Pulso $\leq 5\%$.
- Entorno operativo: temperatura 10 - 40 °C.
- Humedad relativa $\leq 80\%$.

El sensor debe estar conectado en el Jack de presión arterial indicado en la placa de MySignals y funciona con una batería recargable interna. Use el cable específico de presión arterial para cargar el sensor conectado a MySignals.



Fig. 42. Cable de datos y alimentación del sensor.

Tabla 9. Rango de medición.

Parámetro	Unidad	Distancia
Presión sistólica	mm Hg	0-300 mmHg
Presión diastólica	mm Hg	0-300 mmHg
Pulso	ppm	30 ~ 200 ppm

Conecte el sensor en el conector de presión arterial indicado en la placa de hardware de MySignals. El cable del sensor solo tiene una forma de conexión para evitar errores y facilitar la conexión. Use el conector mini-USB para vincular el monitor de presión arterial con la placa MySignals, usando el conector Jack normal (3,5 mm) del cable en este lado.

Antes de comenzar a usar el esfigmomanómetro, se necesita conectar el sensor en la placa MySignals. Después de eso se puede obtener toda la información contenida en el dispositivo.

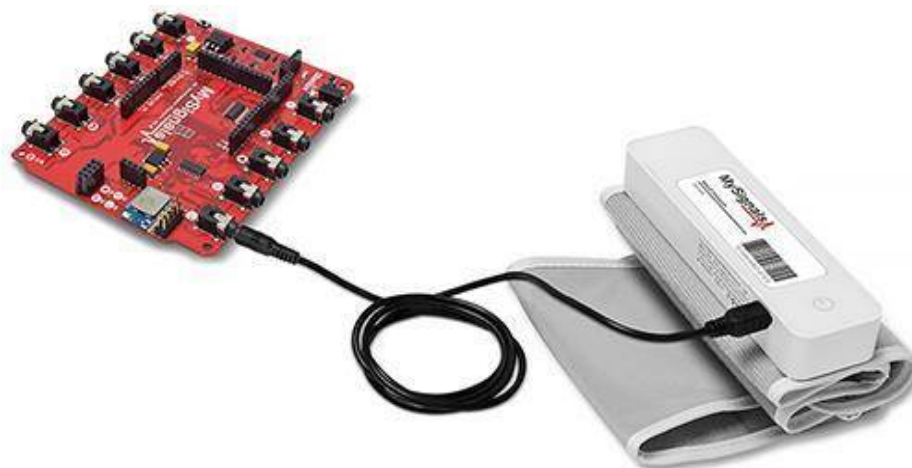


Fig. 43. Conexión del sensor.

Coloque el esfigmomanómetro en su brazo (zona de bíceps) como se muestra en la Fig. 44.

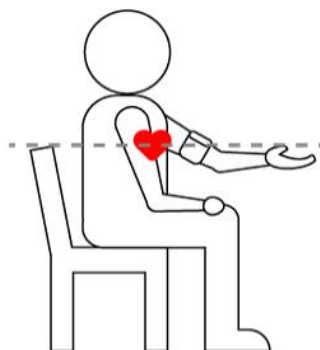


Fig. 44. Posición del sensor para medición.

Encienda el brazalete del esfigmomanómetro (presione el botón ON). El sensor comenzará a realizar una medición. Para medir correctamente es importante mantener el brazo y el brazalete en la posición correcta. Ver Fig. 45.



Fig. 45. Puesta de sensor en brazo.

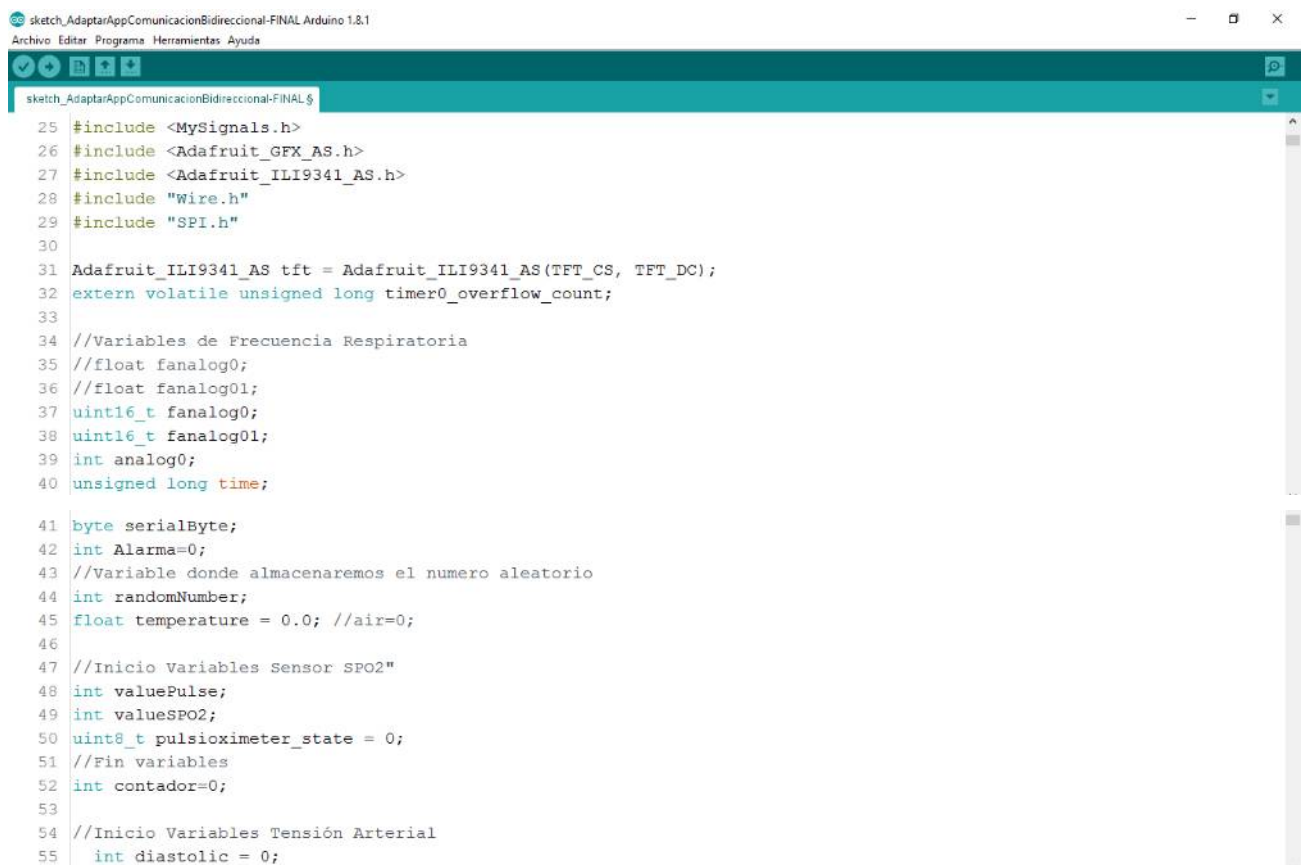
Nota: En el proceso de medición, no hacer movimientos bruscos o la medida no será confiable.

El esfigmomanómetro tardará unos segundos en calcular la lectura de la presión arterial.

Después de unos segundos, obtendrá los valores en la pantalla TFT, Dashboard y pantalla del Smartphone.

Código fuente o Sketch del controlador del sistema.

Se presenta el código fuente generado para el funcionamiento del dispositivo, el cual permite la medición de cualquier variable de las antes mencionadas para este sistema. Se hace mención que gracias al estudio de la tarjeta MySignals (pinout) y la tarjeta controladora (Arduino Mega) se hizo posible el hecho de poder probar cualquier sensor sin tener que volver a reprogramar el controlador por cada prueba de sensor y esto es debido a la poca capacidad de memoria de programa y datos del controlador que el fabricante de MySignals designa para efectuar pruebas básicas. Existen otras investigaciones realizadas como esta, pero se limitan a pruebas con el controlador básico. Ver Fig. 6.



```
sketch_AdaptarAppComunicacionBidireccional-FINAL Arduino 1.8.1
Archivo Editar Programa Herramientas Ayuda
sketch_AdaptarAppComunicacionBidireccional-FINAL.$
25 #include <MySignals.h>
26 #include <Adafruit_GFX_AS.h>
27 #include <Adafruit_ILI9341_AS.h>
28 #include "Wire.h"
29 #include "SPI.h"
30
31 Adafruit_ILI9341_AS tft = Adafruit_ILI9341_AS(TFT_CS, TFT_DC);
32 extern volatile unsigned long timer0_overflow_count;
33
34 //Variables de Frecuencia Respiratoria
35 //float fanalog0;
36 //float fanalog01;
37 uint16_t fanalog0;
38 uint16_t fanalog01;
39 int analog0;
40 unsigned long time;
41
42 byte serialByte;
43 int Alarma=0;
44 //Variable donde almacenaremos el numero aleatorio
45 int randomNumber;
46 float temperature = 0.0; //air=0;
47
48 //Inicio Variables sensor SPO2"
49 int valuePulse;
50 int valueSPO2;
51 uint8_t pulsioximeter_state = 0;
52 //Fin variables
53 int contador=0;
54
55 //Inicio Variables Tensión Arterial
56 int diastolic = 0;
```

```

56 int systolic = 0;
57 int pulseMin = 0;
58 int diastolic1 = 0;
59 int systolic1 = 0;
60 int pulseMin1 = 0;
61 //Fin Variables Tensión Arterial
62
63 //float voltage=0;
64 float voltageAnt=0;
65 float voltageAnt1=0;
66 float gradosF = 0.0;
67
68 int spo2 = 0;
69 int fc = 0;
70 int spo21 = 0;
71
72 int fc1 = 0;
73
74 void setup()
75 {
76 MySignals.begin();
77 tft.init();
78 //tft.setRotation(2); //Vertical
79 tft.setRotation(1); //Horizontal
80
81 //tft.fillScreen(ILI9341_RED);
82 tft.fillScreen(ILI9341_BLACK);
83
84 tft.setTextColor(ILI9341_WHITE);
85 tft.setTextSize(4);
86 //Escribimos el texto:
87
88 // [izq,der], [arriba,abajo]
89 tft.setCursor(15,0);
90 tft.print("Proyecto IoT");
91
92 tft.setTextSize(3);
93 tft.setTextColor(ILI9341_YELLOW);
94 tft.setCursor(30,40);
95 tft.print("Signos Vitales");
96
97 tft.setTextSize(3);
98 tft.setTextColor(ILI9341_GREEN);
99 tft.setCursor(10,100);
100 tft.print("Esperando Comando~App");
101
102 tft.setTextSize(2);
103
104 tft.setTextColor(ILI9341_WHITE);
105 //tft.setCursor(3,200);
106 //tft.setCursor(3,170);
107 tft.setCursor(3,190);
108 tft.print("Investigador: Manuel Gamez");
109 tft.setCursor(3,210);
110 tft.print("Copyright. MEGATEC-HOSPITAL, Zacatecoluca, 2019-2020");
111
112 //Dejamos establecido a un tamaño de texto 8
113 /*tft.setTextSize(5);
114 tft.setTextColor(ILI9341_GREEN);
115 tft.setCursor(200,100);
116 tft.print("oC");*/

```

```

116 //Serial.begin(9600);
117 Serial.begin(19200); //Sensor SpO2 Oximetria funciona a 2 velocidades: 115200 y 19200.
118 //Los demás sensores: Temperatura Corporal, Frecuencia Respiratoria funcionan a varias.
119 //El sensor de tensión arterial solo funciona a velocidad de 19200 baudios y debido a
120 //esto adapté el sketch para que todos los sensores trabajen a esta velocidad...
121 //Serial.begin(115200); //A esta velocidad trabajan bien los 2 sensores UART
122 //Oximetria y Tensión Arterial.
123 //OJO: El sensor de Tensión Arterial solo funciona a velocidad de 115200 baudios.
124 // por tal razón hay que trabajar a esta ya que tambien lo permiten los demás sensores.
125 Serial1.begin(9600); //Canal Serial de Comunicación: Módulo Bluetooth HC-05.
126 Serial2.begin(9600); //Canal Serial de Comunicación: Módulo FTDI = USB TIPO A MACHO.
127
128 randomSeed(500);
129
130 //Var SPO2
131
132 MySignals.initSensorUART();
133 MySignals.enableSensorUART(PULSIOXIMETER);
134 //Fin Var SpO2
135
136 //Var Tensión o Presión Arterial
137 MySignals.initSensorUART();
138 MySignals.enableSensorUART(BLOODPRESSURE);
139 //Fin Var Tensión Arterial
140
141 Serial2.println("Starting...");
142 } //End function setup

```

```

---
143 void limpiar()
144 {
145 //tft.fillScreen(ILI9341_RED); //Limpio
146 tft.setTextSize(3);
147 tft.setTextColor(ILI9341_BLUE);
148 tft.setCursor(30,40);
149 tft.print("Signos Vitales");
150
151 tft.setTextSize(3);
152 tft.setTextColor(ILI9341_GREEN);
153 tft.setCursor(10,100);
154 tft.print("Esperando Comando~App");
155 }

```

```

156 void LimpioTextoScreen(){
157 //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fondo
158 //y resalto solo el último dato a mostrar o imprimir en pantalla.
159 tft.setTextSize(3);
160 //tft.setTextColor(ILI9341_RED);
161 tft.setTextColor(ILI9341_BLACK);
162 tft.setCursor(30,40);
163 tft.print("Signos Vitales");
164
165 tft.setTextSize(3);
166 //tft.setTextColor(ILI9341_RED);
167 tft.setTextColor(ILI9341_BLACK);
168 tft.setCursor(10,100);
169 tft.print("Esperando Comando~App");
170 //End del bloque
171 }

```

```

172
173
174 void loop()
175 {
176 Alarma=1;
177 //Genera un numero aleatorio entre 1 y 1023
178 //randomNumber = random(1,1023);
179
180 while (Serial1.available()>0)
181 {
182 serialByte = Serial1.read();
183 if (serialByte == 'R')
184 {
185 while(1)
186 {
187 //Sensor para variable biométrica AirFlow o Flujo de Aire

```

```

188 //uint16_t air = (uint16_t)MySignals.getAirflow(DATA);
189 //Serial.println("Flujo de Aire: ");
190 //Serial.println(air);
191 //delay(20);
192
193 /*uint16_t air = (uint16_t)MySignals.getAirflow(DATA);
194 Serial.println(air);
195 delay(20);*/
196
197 randomNumber = random(1,1023);
198
199 /*
200 * uint16_t air = (uint16_t)MySignals.getAirflow(DATA);
201 * Serial.println(air);
202 * delay(20);
203
204 */
205 //uint16_t air = (uint16_t)MySignals.getAirflow(DATA);
206 fanalog0 = (uint16_t)MySignals.getAirflow(DATA);
207
208 if(contador==0){
209 //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fondo
210 //y resalto solo el último dato a mostrar o imprimir en pantalla.
211 tft.setTextSize(3);
212 //tft.setTextColor(ILI9341_RED);
213 tft.setTextColor(ILI9341_BLACK);
214 tft.setCursor(30,40);
215 tft.print("Signos Vitales");
216
217 tft.setTextSize(3);
218
219 //tft.setTextColor(ILI9341_RED);
220 tft.setTextColor(ILI9341_BLACK);
221 tft.setCursor(10,100);
222 tft.print("Esperando Comando~App");
223 //End del bloque
224 contador=1;
225 }
226
227 if(fanalog0-fanalog01>0.01||fanalog0-fanalog01<-0.01)
228 {
229 tft.setTextSize(8);
230 tft.setTextColor(ILI9341_BLACK);
231 tft.setCursor(30,100);
232 tft.print(fanalog01);
233
234 tft.setTextColor(ILI9341_GREEN);
235 tft.setCursor(30,100);
236 tft.print(fanalog0);
237 fanalog01=fanalog0;
238
239 tft.setTextSize(2);
240 tft.setTextColor(ILI9341_YELLOW);
241 tft.setCursor(20,40);
242 tft.print("<<<FREC. RESPIRATORIA>>>");
243 }
244
245 // Use the timer0 => 1 tick every 4 us
246 //time = (timer0_overflow_count << 8) + TCNT0;
247 // Microseconds conversion.

```



```

248 //time = time*4;
249
250 //Print in a file for simulation
251 /*
252 Serial1.print(time);
253 Serial1.print(",");
254 Serial1.println(fanalog0,5);
255 */
256
257 Serial1.print(0); //SpO2 = Saturación Parcial del Oxígeno y Pulso/Frecuencia Cardiaca.
258 Serial1.print(",");
259 Serial1.print(randomNumber); //Frec_cardiaca
260 Serial1.print(",");
261 /* Inicio Tension Arterial */
262 //Serial1.print(0); //Tensión Arterial
263
264 Serial1.print(0); //Diastolic
265 Serial1.print(","); //Separador
266 Serial1.print(0); //Systolic
267 Serial1.print(","); //Separador
268 Serial1.print(0); //Pulse / min
269 /* Fin Tensión Arterial */
270 Serial1.print(",");
271 Serial1.print(fanalog0); //Frecuencia Respiratoria
272 Serial1.print(",");
273 Serial1.print(0); //Temperatura Corporal
274 Serial1.print(",");
275 Serial1.print(Alarma); //Alarma
276 Serial1.print(",");
277 Serial1.println("~");
278 Serial1.flush();
279
280 Serial2.print(0); //SpO2 = Saturación Parcial del Oxígeno y Pulso/Frecuencia Cardiaca.
281 Serial2.print(",");
282 Serial2.print(randomNumber); //Frec_cardiaca
283 Serial2.print(",");
284 /* Inicio Tension Arterial */
285 //Serial2.print(0); //Tensión Arterial
286 Serial2.print(0); //Diastolic
287 Serial2.print(","); //Separador
288 Serial2.print(0); //Systolic
289 Serial2.print(","); //Separador
290 Serial2.print(0); //Pulse / min
291 /* Fin Tensión Arterial */
292 Serial2.print(",");
293 Serial2.print(fanalog0); //Frecuencia Respiratoria
294 Serial2.print(",");
295 Serial2.print(0); //Temperatura Corporal
296
297 Serial2.print(",");
298 Serial2.print(Alarma); //Alarma
299 Serial2.print(",");
300 Serial2.println("~");
301 Serial2.flush();
302
303 if (Serial1.available()>0)
304 {
305 serialByte = Serial1.read();
306 //if (serialByte == 'F') break;
307 if (serialByte == 'F'){
308 tft.setTextSize(8);
309 tft.setTextColor(ILI9341_BLACK);
310 tft.setCursor(30,100);

```

```

311         tft.print(fanalog01);
312
313         tft.setTextColor(ILI9341_BLACK);
314         tft.setCursor(30,100);
315         tft.print(fanalog0);
316         //fanalog01=fanalog0;
317
318         tft.setTextSize(2);
319         tft.setTextColor(ILI9341_BLACK);
320         tft.setCursor(20,40);
321         tft.print("<<<FREC. RESPIRATORIA>>>");
322
323         //Ahora mandare a pantalla el texto predeterminado de inicio.
324         if(contador==1){
325             //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fo
326
327             //y resalto solo el último dato a mostrar o imprimir en pantalla.
328             tft.setTextSize(3);
329             //tft.setTextColor(ILI9341_RED);
330             tft.setTextColor(ILI9341_YELLOW);
331             tft.setCursor(30,40);
332             tft.print("Signos Vitales");
333
334             tft.setTextSize(3);
335             //tft.setTextColor(ILI9341_RED);
336             tft.setTextColor(ILI9341_GREEN);
337             tft.setCursor(10,100);
338             tft.print("Esperando Comando~App");
339             contador=0;
340         }
341         //End del bloque
342
343         break;
344     }
345     }
346     }
347
348     //Sensor de Temperatura Corporal.
349     //serialByte = Serial1.read();
350     if (serialByte == 'T')
351     {
352         while(1)
353         {
354             randomNumber = random(1,1023);
355             float temperature = MySignals.getCalibratedTemperature(100, 10, -2, 5);
356
357             gradosF = ((temperature * 1.8)+32.0);
358
359             if(contador==0){
360                 //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fondo
361                 //y resalto solo el último dato a mostrar o imprimir en pantalla.
362                 tft.setTextSize(3);
363                 //tft.setTextColor(ILI9341_RED);
364                 tft.setTextColor(ILI9341_BLACK);
365                 tft.setCursor(30,40);
366                 tft.print("Signos Vitales");
367
368                 tft.setTextSize(3);
369                 //tft.setTextColor(ILI9341_RED);
370                 tft.setTextColor(ILI9341_BLACK);
371                 tft.setCursor(10,100);

```

```

371     tft.print("Esperando Comando~App");
372     //End del bloque
373     contador=1;
374 }
375 if(temperature-voltageAnt>0.01||temperature-voltageAnt<-0.01)
376 {
377     tft.setTextSize(5);
378     //tft.setTextColor(IL19341_RED);
379     tft.setTextColor(IL19341_BLACK); //limpiar
380     tft.setCursor(30,80);
381     tft.print(voltageAnt);
382     tft.print(" oc");
383
384     tft.setTextSize(5);
385     //tft.setTextColor(IL19341_RED);
386
387     tft.setTextColor(IL19341_BLACK); //limpiar
388     tft.setCursor(30,140);
389     tft.print(voltageAnt1, 2);
390     tft.print(" oF");
391
392     //delay(10);
393     //tft.drawNumber(voltageAnt, 100, 30, 5);
394     //tft.drawNumber(voltageAnt, 0, 30, 2);
395     tft.setTextColor(IL19341_GREEN);
396     tft.setCursor(30,80);
397     tft.print(temperature);
398     tft.print(" oc");
399     voltageAnt=temperature;
400
401     tft.setTextColor(IL19341_GREEN);
402
403     tft.setCursor(30,140);
404     tft.print(gradesF, 2);
405     tft.print(" oF");
406     voltageAnt1=gradesF;
407
408     tft.setTextSize(3);
409     tft.setTextColor(IL19341_YELLOW);
410     tft.setCursor(20,40);
411     tft.print("<<TEMPERATURA>>");
412 }
413
414 //Envio los datos de sensores por Bluetooth
415 Serial1.print(0); //SpO2 = Saturación Parcial del Oxígeno y Pulso/Frecuencia Cardiaca.
416 Serial1.print(",");
417 Serial1.print(randomNumber); //Frec_cardiaca
418
419 /* Inicio Tension Arterial */
420 Serial1.print(0); //Tensión Arterial
421 Serial1.print(0); //Diastolic
422 Serial1.print(","); //Separador
423 Serial1.print(0); //Systolic
424 Serial1.print(","); //Separador
425 Serial1.print(0); //Pulse / min
426 /* Fin Tensión Arterial */
427 Serial1.print(",");
428 Serial1.print(0); //Frecuencia Respiratoria
429 Serial1.print(",");
430 Serial1.print(temperature, 2); //Temperatura Corporal
431 Serial1.print(",");
432 Serial1.print(Alarma); //Alarma

```

```

431 Serial1.print(",");
432 Serial1.println("~");
433 Serial1.flush();
434
435 //Envio los datos de sensores por interface FTDI a USB
436 Serial2.print(0); //SpO2 = Saturación Parcial del Oxígeno y Pulso/Frecuencia Cardiaca.
437 Serial2.print(",");
438 Serial2.print(randomNumber); //Frec_cardiaca;
439 Serial2.print(",");
440 /* Inicio Tension Arterial */
441 //Serial1.print(0); //Tensión Arterial
442 Serial2.print(0); //Diastolic
443 Serial2.print(","); //Separador
444 Serial2.print(0); //Systolic
445 Serial2.print(","); //Separador
446
447 Serial2.print(0); //Pulse / min
448 /* Fin Tensión Arterial */
449 Serial2.print(",");
450 Serial2.print(","); //Frecuencia Respiratoria
451 Serial2.print(temperature, 2); //Temperatura Corporal
452 Serial2.print(",");
453 Serial2.print(Alarma); //Alarma
454 Serial2.print(",");
455 Serial2.println("~");
456 Serial2.flush();
457
458 if (Serial1.available()>0)
459 {
460     serialByte = Serial1.read();
461
462     //if (serialByte == 'C') break;
463     if (serialByte == 'C'){
464         tft.setTextSize(5);
465         //tft.setTextColor(IL19341_RED);
466         tft.setTextColor(IL19341_BLACK); //limpiar
467         tft.setCursor(30,80);
468         tft.print(voltageAnt);
469         tft.print(" oC");
470
471         tft.setTextSize(5);
472         //tft.setTextColor(IL19341_RED);
473         tft.setTextColor(IL19341_BLACK); //limpiar
474         tft.setCursor(30,140);
475         tft.print(voltageAnt1, 2);
476         tft.print(" oF");
477
478         tft.setTextColor(IL19341_BLACK);
479         tft.setCursor(30,80);
480         tft.print(temperature);
481         tft.print(" oC");
482         voltageAnt=temperature;
483
484         tft.setTextColor(IL19341_BLACK);
485         tft.setCursor(30,140);
486         tft.print(gradosF, 2);
487         tft.print(" oF");
488         voltageAnt1=gradosF;
489
490         tft.setTextSize(3);
491         tft.setTextColor(IL19341_BLACK);

```

```

491     tft.setCursor(20,40);
492     tft.print("<<TEMPERATURA>>");
493     //Hasta aqui termino de limpiar.
494
495     //Ahora mandare a pantalla el texto predeterminado de inicio.
496     if(contador==1){
497         //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fondo
498         //y resalto solo el último dato a mostrar o imprimir en pantalla.
499         tft.setTextSize(3);
500         //tft.setTextColor(ILI9341_RED);
501         tft.setTextColor(ILI9341_YELLOW);
502         tft.setCursor(30,40);
503         tft.print("Signos Vitales");
504
505         tft.setTextSize(3);
506
507         //tft.setTextColor(ILI9341_RED);
508         tft.setTextColor(ILI9341_GREEN);
509         tft.setCursor(10,100);
510         tft.print("Esperando Comando-App");
511         contador=0;
512     }
513     //End del bloque
514     break;
515 }
516 delay(1000); //
517 }
518 }
519
520 //Sensor Oximetria = Sensor de pulso y oxigeno en la sangre.
521
522 //Sensor Oximetria = Frecuencia Cardiaca y SpO2 (Saturación Parcial del Oxígeno).
523 //serialByte = Serial1.read();
524 if (serialByte == '0')
525 {
526     while(1)
527     {
528         //BEGIN SPO2
529         if (MySignals.spo2_micro_detected == 0 && MySignals.spo2_mini_detected == 0){
530             uint8_t statusPulsioximeter = MySignals.getStatusPulsioximeterGeneral();
531
532             if (statusPulsioximeter == 1){
533                 MySignals.spo2_mini_detected = 0;
534                 MySignals.spo2_micro_detected = 1;
535                 //tft.drawString("Micro detected", 0, 100, 2); //Mi conector de sensores....
536             }else if (statusPulsioximeter == 2){
537                 MySignals.spo2_mini_detected = 1;
538                 MySignals.spo2_micro_detected = 0;
539                 //tft.drawString("Mini detected", 0, 100, 2);
540             }else{
541                 MySignals.spo2_micro_detected = 0;
542                 MySignals.spo2_mini_detected = 0;
543             }
544         }
545     }
546     if(MySignals.spo2_micro_detected == 1){
547         MySignals.enableSensorUART(PULSIOXIMETER_MICRO);
548         delay(10);
549         uint8_t getPulsioximeterMicro_state = MySignals.getPulsioximeterMicro();
550         if (getPulsioximeterMicro_state == 1){
551             /*

```

```

551 tft.drawNumber(MySignals.pulsioximeterData.BPM, 0, 60, 2);
552 tft.drawString("PRbpm", 25, 65, 2);
553 tft.drawNumber(MySignals.pulsioximeterData.O2, 0, 75, 2);
554 tft.drawString("%SpO2", 25, 80, 2);
555 */
556
557 if(contador==0){
558     //Begin Bloque:Con este bloque limpio el texto o lo ocultó pintandole del mismo color del fondo
559     //y resalto solo el último dato a mostrar o imprimir en pantalla.
560     tft.setTextSize(3);
561     //tft.setTextColor(IL19341_RED);
562     tft.setTextColor(IL19341_BLACK);
563     tft.setCursor(30,40);
564     tft.print("Signos Vitales");
565
566     tft.setTextSize(3);
567     //tft.setTextColor(IL19341_RED);
568     tft.setTextColor(IL19341_BLACK);
569     tft.setCursor(10,100);
570     tft.print("Esperando Comando~App");
571     //End del bloque
572     contador=1;
573 }
574
575 spo2 = MySignals.pulsioximeterData.O2;
576 fc = MySignals.pulsioximeterData.BPM;
577
578 if(spo2-spo21>0.01||spo2-spo21<-0.01||fc-fc1>0.01||fc-fc1<-0.01)
579 {
580     tft.setTextSize(5);
581
582     //tft.setTextColor(IL19341_RED);
583     tft.setTextColor(IL19341_BLACK); //limpiar
584     tft.setCursor(30,80);
585     tft.print(spo2);
586     tft.print(" %");
587
588     tft.setTextSize(5);
589     //tft.setTextColor(IL19341_RED);
590     tft.setTextColor(IL19341_BLACK); //limpiar
591     tft.setCursor(30,80);
592     tft.print(spo21);
593     tft.print(" %");
594
595     tft.setTextSize(5);
596     //tft.setTextColor(IL19341_RED);
597
598     tft.setTextColor(IL19341_BLACK); //limpiar
599     tft.setCursor(30,140);
600     tft.print(fc);
601     tft.print(" PRbpm");
602
603     tft.setTextSize(5);
604     //tft.setTextColor(IL19341_RED);
605     tft.setTextColor(IL19341_BLACK); //limpiar
606     tft.setCursor(30,140);
607     tft.print(fc1);
608     tft.print(" PRbpm");
609
610     //delay(100);
611
612     tft.setTextColor(IL19341_GREEN);

```

```

611         tft.setTextSize(5);
612         tft.setCursor(30,80);
613         tft.print(spo2);
614         tft.print(" %");
615         spo21=spo2;
616
617         //tft.setTextColor(ILI9341_GREEN);
618         tft.setTextSize(5);
619         tft.setCursor(30,140);
620         tft.print(fc);
621         tft.print(" PRbpm");
622         fcl=fc;
623
624         tft.setTextSize(3);
625         tft.setTextColor(ILI9341_YELLOW);
626
627         tft.setCursor(30,40);
628         tft.print("<<<OXIMETRIA>>>");
629     }
630
631     randomNumber = random(1,1023);
632     //float temperature = MySignals.getCalibratedTemperature(100, 10, -2, 5);
633     //Envio los datos de sensores por Bluetooth
634     Serial1.print(spo2); //SpO2 = Saturación Parcial del Oxígeno y Pulso/Frecuencia Cardiaca.
635     Serial1.print(",");
636     Serial1.print(fc); //Frec_cardiaca;
637     Serial1.print(",");
638     /* Inicio Tension Arterial */
639     //Serial1.print(0); //Tensión Arterial
640     Serial1.print(0); //Diastolic
641     Serial1.print(","); //Separador
642
643     Serial1.print(0); //Systolic
644     Serial1.print(","); //Separador
645     Serial1.print(0); //Pulse / min
646     /* Fin Tensión Arterial */
647     Serial1.print(",");
648     Serial1.print(0); //Frecuencia Respiratoria
649     Serial1.print(",");
650     Serial1.print(0); //Temperatura Corporal
651     Serial1.print(",");
652     Serial1.print(Alarma); //Alarma
653     Serial1.print(",");
654     Serial1.println("~");
655     Serial1.flush();
656
657     //Envio los datos de sensores por interface FTDI a USB
658
659     Serial2.print(MySignals.pulsioximeterData.O2); //SpO2 = Saturación Parcial del Oxígeno
660     Serial2.print(",");
661     Serial2.print(MySignals.pulsioximeterData.BPM); //Serial.print(Frec_cardiaca);
662     Serial2.print(",");
663     /* Inicio Tension Arterial */
664     //Serial1.print(0); //Tensión Arterial
665     Serial2.print(0); //Diastolic
666     Serial2.print(","); //Separador
667     Serial2.print(0); //Systolic
668     Serial2.print(","); //Separador
669     Serial2.print(0); //Pulse / min
670     /* Fin Tensión Arterial */
671     Serial2.print(",");
672     Serial2.print(0); //Frecuencia Respiratoria
673     Serial2.print(",");

```

```

671     Serial2.print(0);           //Temperatura Corporal
672     Serial2.print(",");
673     Serial2.print(Alarma);     //Alarma
674     Serial2.print(",");
675     Serial2.println("~");
676     Serial2.flush();
677
678     }else if (getPulsioximeterMicro_state == 2){
679
680         limpioDataDesconexionSensorSpO2();
681
682         //Serial.println(F("Finger out or calculating"));
683         Serial2.println(F("X"));           //Perdiendo la captura de datos del sensor. Se quito el dedo...
684
685         //Serial1.println(F("0"));           //Mando centinela a la App para saber que se ha quitado el dedo del
686
687         Serial1.print(0);           //SpO2 = Saturación Parcial del Oxígeno y Pulso/Frecuencia Cardiaca.
688         Serial1.print(",");
689         Serial1.print(0);           //Frec_cardiaca
690         Serial1.print(",");
691         /* Inicio Tension Arterial */
692         //Serial1.print(0);           //Tensión Arterial
693         Serial1.print(0);           //Diastolic
694         Serial1.print(",");         //Separador
695         Serial1.print(0);           //Systolic
696         Serial1.print(",");         //Separador
697         Serial1.print(0);           //Pulse / min
698         /* Fin Tensión Arterial */
699         Serial1.print(",");
700         Serial1.print(0);           //Frecuencia Respiratoria
701
702         Serial1.print(0);           //Temperatura Corporal
703         Serial1.print(",");
704         Serial1.print(Alarma);     //Alarma
705         Serial1.print(",");
706         Serial1.println("~");
707         Serial1.flush();
708     }else{
709         MySignals.spo2_micro_detected = 0;
710         //Serial.println(F("SPO2 Micro lost connection"));
711         Serial2.println(F("Y"));           //Se perdió por completo la captura de datos.
712     }
713 }else{
714
715     limpioDataDesconexionSensorSpO2();
716
717     Serial2.println(F("Error de Lectura de datos: ..."));
718     //Serial2.println(F("1. El sensor de oximetría esta desconectado de la tarjeta MySignals"));
719     //Serial2.println(F("2. El sensor de oximetría esta conectado pero no ha colocado su dedo indice en el.
720     //Serial1.println(F("0"));           //Mando centinela a la App para saber que se ha quitado el dedo del se
721     Serial1.print(0);           //SpO2 = Saturación Parcial del Oxígeno y Pulso/Frecuencia Cardiaca.
722     Serial1.print(",");
723     Serial1.print(0);           //Frec_cardiaca
724     Serial1.print(",");
725     /* Inicio Tension Arterial */
726     //Serial1.print(0);           //Tensión Arterial
727     Serial1.print(0);           //Diastolic
728     Serial1.print(",");         //Separador
729     Serial1.print(0);           //Systolic
730     Serial1.print(",");         //Separador
731     Serial1.print(0);           //Pulse / min

```



```

731     /* Fin Tensión Arterial */
732     Serial1.print(",");
733     Serial1.print(0);           //Frecuencia Respiratoria
734     Serial1.print(",");
735     Serial1.print(0);           //Temperatura Corporal
736     Serial1.print(",");
737     Serial1.print(Alarma);      //Alarma
738     Serial1.print(",");
739     Serial1.println("~");
740     Serial1.flush();
741 }
742 //END SPO2
743
744 if (Serial1.available()>0)
745 {
746     serialByte = Serial1.read();
747     //if (serialByte == 'S') break;
748     if (serialByte == 'S'){
749         tft.setTextColor(IL19341_BLACK);
750         tft.setTextSize(5);
751         tft.setCursor(30,80);
752         tft.print(spo2);
753         tft.print(" %");
754         //spo21=spo2;
755
756         tft.setTextColor(IL19341_BLACK);
757         tft.setTextSize(5);
758         tft.setCursor(30,80);
759         tft.print(spo21);
760         tft.print(" %");
761
762         //spo21=spo2;
763
764         tft.setTextColor(IL19341_BLACK);
765         tft.setTextSize(5);
766         tft.setCursor(30,140);
767         tft.print(fc);
768         tft.print(" PRbpm");
769         //fc1=fc;
770
771         tft.setTextColor(IL19341_BLACK);
772         tft.setTextSize(5);
773         tft.setCursor(30,140);
774         tft.print(fc1);
775         tft.print(" PRbpm");
776         //fc1=fc;
777
778         tft.setTextSize(3);
779         tft.setTextColor(IL19341_BLACK);
780         tft.setCursor(30,40);
781         tft.print("<<<OXIMETRIA>>>");
782
783         //Ahora mandare a pantalla el texto predeterminado de inicio.
784         //if(contador==1){
785
786             //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fondo
787             //y resalto solo el último dato a mostrar o imprimir en pantalla.
788             tft.setTextSize(3);
789             //tft.setTextColor(IL19341_RED);
790             tft.setTextColor(IL19341_YELLOW);
791             tft.setCursor(30,40);

```

```

791         tft.print("Signos Vitales");
792
793         tft.setTextSize(3);
794         //tft.setTextColor(ILI9341_RED);
795         tft.setTextColor(ILI9341_GREEN);
796         tft.setCursor(10,100);
797         tft.print("Esperando Comando-App");
798         contador=0;
799         //)
800         //End del bloque
801         break;
802     }
803 }
804
805     delay(1000);
806
807 }
808
809 //Sensor Tensión/Presión Arterial.
810 //serialByte = Serial1.read();
811 if (serialByte == 'P')
812 {
813     while(1)
814     {
815         //Begin Tensión o Presión Arterial
816         if (MySignals.getStatusBP())
817         {
818             delay(1000);
819             if (MySignals.getBloodPressure() == 1)
820             {
821
822                 MySignals.disableMuxUART();
823                 /*
824                 Serial2.println();
825                 Serial2.print("Diastolic: ");
826                 Serial2.println(MySignals.bloodPressureData.diastolic);
827                 Serial2.print("Systolic: ");
828                 Serial2.println(MySignals.bloodPressureData.systolic);
829                 Serial2.print("Pulse/min: ");
830                 Serial2.println(MySignals.bloodPressureData.pulse);
831                 */
832
833                 if(contador==0){
834                     //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fondo
835                     //y resalto solo el último dato a mostrar o imprimir en pantalla.
836                     tft.setTextSize(3);
837
838                     //tft.setTextColor(ILI9341_RED);
839                     tft.setTextColor(ILI9341_BLACK);
840                     tft.setCursor(30,40);
841                     tft.print("Signos Vitales");
842
843                     tft.setTextSize(3);
844                     //tft.setTextColor(ILI9341_RED);
845                     tft.setTextColor(ILI9341_BLACK);
846                     tft.setCursor(10,100);
847                     tft.print("Esperando Comando-App");
848                     //End del bloque
849                     contador=1;
850                 }
851
852                 //diastolic,systolic,pulseMin,diastolic1,systolic1,pulseMin1;

```

```

851     diastolic = MySignals.bloodPressureData.diastolic;
852     systolic = MySignals.bloodPressureData.systolic;
853     pulseMin = MySignals.bloodPressureData.pulse;
854
855     if(diastolic-diastolic1>0.01||diastolic-diastolic1<-0.01||systolic-systolic1>0.01||systolic-systolic1<-
856     {
857         //Oculto información.
858         tft.setTextSize(4);
859         //tft.setTextColor(IL19341_RED);
860         tft.setTextColor(IL19341_BLACK); //limpiar
861         tft.setCursor(30,80);
862         tft.print(diastolic);
863         tft.print(" mmHg");
864
865         tft.setTextSize(4);
866
867         //tft.setTextColor(IL19341_RED);
868         tft.setTextColor(IL19341_BLACK); //limpiar
869         tft.setCursor(30,80);
870         tft.print(diastolic1);
871         tft.print(" mmHg");
872
873         tft.setTextSize(4);
874         //tft.setTextColor(IL19341_RED);
875         tft.setTextColor(IL19341_BLACK); //limpiar
876         tft.setCursor(30,120);
877         tft.print(systolic);
878         tft.print(" mmHg");
879
880         tft.setTextSize(4);
881         //tft.setTextColor(IL19341_RED);
882
883         tft.setTextColor(IL19341_BLACK); //limpiar
884         tft.setCursor(30,120);
885         tft.print(systolic1);
886         tft.print(" mmHg");
887
888         tft.setTextSize(4);
889         //tft.setTextColor(IL19341_RED);
890         tft.setTextColor(IL19341_BLACK); //limpiar
891         tft.setCursor(30,160);
892         tft.print(pulseMin);
893         tft.print(" bpm");
894
895         tft.setTextSize(4);
896         //tft.setTextColor(IL19341_RED);
897         tft.setTextColor(IL19341_BLACK); //limpiar
898
899         tft.setCursor(30,160);
900         tft.print(pulseMin1);
901         tft.print(" bpm");
902
903         //Muestro data en pantalla.
904         tft.setTextColor(IL19341_GREEN);
905         tft.setTextSize(4);
906         tft.setCursor(30,80);
907         tft.print(diastolic);
908         tft.print(" mmHg");
909         diastolic1=diastolic;
910
911         tft.setTextColor(IL19341_GREEN);
912         tft.setTextSize(4);
913         tft.setCursor(30,120);

```

```

911         tft.print(systolic);
912         tft.print(" mmHg");
913         systolic1=systolic;
914
915         tft.setTextColor(ILI9341_GREEN);
916         tft.setTextSize(4);
917         tft.setCursor(30,160);
918         tft.print(pulseMin);
919         tft.print(" bpm");
920         pulseMin1=pulseMin;
921
922         tft.setTextSize(3);
923         tft.setTextColor(ILI9341_YELLOW);
924         tft.setCursor(10,40);
925         tft.print("Tension Arterial");
926     }
927
928     randomNumber = random(1,1023);
929     //float temperature = MySignals.getCalibratedTemperature(100, 10, -2, 5);
930     //Envio los datos de sensores por Bluetooth
931     Serial1.print(0); //SpO2 = Saturación Parcial del Oxígeno y Pulso/Fr
932     Serial1.print(",");
933     Serial1.print(0); //Frec_cardiaca;
934     Serial1.print(",");
935     /* Inicio Tension Arterial */
936     //Serial1.print(0); //Tensión Arterial
937     Serial1.print(diastolic); //Diastolic
938     Serial1.print(","); //Separador
939     Serial1.print(systolic); //Systolic
940     Serial1.print(","); //Separador
941
942     Serial1.print(pulseMin); //Pulse / min
943     /* Fin Tensión Arterial */
944     Serial1.print(",");
945     Serial1.print(0); //Frecuencia Respiratoria
946     Serial1.print(0); //Temperatura Corporal
947     Serial1.print(",");
948     Serial1.print(Alarma); //Alarma
949     Serial1.print(",");
950     Serial1.println("~");
951     Serial1.flush();
952
953     //Envio los datos de sensores por interface FTDI a USB
954     Serial2.print(0); //SpO2 = Saturación Parcial del Oxígeno y Pulso/1
955     Serial2.print(",");
956
957     Serial2.print(0); //Serial.print(Frec_cardiaca);
958     Serial2.print(",");
959     /* Inicio Tension Arterial */
960     //Serial1.print(0); //Tensión Arterial
961     Serial2.print(MySignals.bloodPressureData.diastolic); //Diastolic
962     Serial2.print(","); //Separador
963     Serial2.print(MySignals.bloodPressureData.systolic); //Systolic
964     Serial2.print(","); //Separador
965     Serial2.print(MySignals.bloodPressureData.pulse); //Pulse / min
966     /* Fin Tensión Arterial */
967     Serial2.print(",");
968     Serial2.print(0); //Frecuencia Respiratoria
969     Serial2.print(",");
970     Serial2.print(0); //Temperatura Corporal
971     Serial2.print(",");

```

```

971     Serial2.print(Alarma);           //Alarma
972     Serial2.print(",");
973     Serial2.println("~");
974     Serial2.flush();
975
976     MySignals.enableMuxUART();
977 }
978 }else{
979     //Aca haré lo mismo que hice con el sensor de oximetria.
980     //limpioDataDesconexionSensorTA();
981
982     //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fondo
983     //y resalto solo el último dato a mostrar o imprimir en pantalla.
984     tft.setTextSize(3);
985     //tft.setTextColor(ILI9341_RED);
986
987     tft.setTextColor(ILI9341_BLACK);
988     tft.setCursor(30,40);
989     tft.print("Signos Vitales");
990
991     tft.setTextSize(3);
992     //tft.setTextColor(ILI9341_RED);
993     tft.setTextColor(ILI9341_BLACK);
994     tft.setCursor(10,100);
995     tft.print("Esperando Comando~App");
996     //End del bloque
997
998     tft.setTextSize(3);
999     tft.setTextColor(ILI9341_YELLOW);
1000    tft.setCursor(10,40);
1001    tft.print("**Tension Arterial");
1002
1003    Serial2.println();
1004    Serial2.println("Error: Sensor desconectado o apagado."); //Mando Mensaje por Interfaz FTDI.
1005    //Serial1.print("0"); //Este va ser el comando para enviar a la App via Bluetooth.
1006    //Servirá para que la App muestre un dialogo que informe al usuario que
1007    //Sensor esta desconectado de la tarjeta MySignals o apagado.
1008    //Envio notificación a la App para que detecte que el sensor esta apagado o desconectado
1009    //de la tarjeta MySignals.
1010    Serial1.print(0); //SpO2 = Saturación Parcial del Oxígeno y Pulso/Frecuencia Cardiaca.
1011    Serial1.print(",");
1012    Serial1.print(0); //Frec_cardiaca
1013    Serial1.print(",");
1014    /* Inicio Tension Arterial */
1015    //Serial1.print(0); //Tensión Arterial
1016    Serial1.print(0); //Diastolic
1017
1018    Serial1.print(","); //Separador
1019    Serial1.print(0); //systolic
1020    Serial1.print(","); //Separador
1021    Serial1.print(0); //Pulse / min
1022    /* Fin Tensión Arterial */
1023    Serial1.print(",");
1024    Serial1.print(0); //Frecuencia Respiratoria
1025    Serial1.print(0); //Temperatura Corporal
1026    Serial1.print(",");
1027    Serial1.print(Alarma); //Alarma
1028    Serial1.print(",");
1029    Serial1.println("~");
1030    Serial1.flush();
1031 }

```

```

1031         //delay(1000);
1032         //End Tensión o Presión Arterial
1033
1034     if (Serial1.available()>0)
1035     {
1036         serialByte = Serial1.read();
1037         //if (serialByte == 'A') break;
1038         if (serialByte == 'A'){
1039             tft.setTextSize(4);
1040             //tft.setTextColor(IL19341_RED);
1041             tft.setTextColor(IL19341_BLACK); //limpiar
1042             tft.setCursor(30,80);
1043             tft.print(diastolic);
1044             tft.print(" mmHg");
1045
1046
1047             tft.setTextSize(4);
1048             //tft.setTextColor(IL19341_RED);
1049             tft.setTextColor(IL19341_BLACK); //limpiar
1050             tft.setCursor(30,80);
1051             tft.print(diastolicl);
1052             tft.print(" mmHg");
1053
1054             tft.setTextSize(4);
1055             //tft.setTextColor(IL19341_RED);
1056             tft.setTextColor(IL19341_BLACK); //limpiar
1057             tft.setCursor(30,120);
1058             tft.print(systolic);
1059             tft.print(" mmHg");
1060
1061             tft.setTextSize(4);
1062             //tft.setTextColor(IL19341_RED);
1063             tft.setTextColor(IL19341_BLACK); //limpiar
1064             tft.setCursor(30,120);
1065             tft.print(systolicl);
1066             tft.print(" mmHg");
1067
1068             tft.setTextSize(4);
1069             //tft.setTextColor(IL19341_RED);
1070             tft.setTextColor(IL19341_BLACK); //limpiar
1071             tft.setCursor(30,160);
1072             tft.print(pulseMin);
1073             tft.print(" bpm");
1074
1075             tft.setTextSize(4);
1076             //tft.setTextColor(IL19341_RED);
1077
1078             tft.setTextColor(IL19341_BLACK); //limpiar
1079             tft.setCursor(30,160);
1080             tft.print(pulseMinl);
1081             tft.print(" bpm");
1082
1083             tft.setTextSize(3);
1084             tft.setTextColor(IL19341_BLACK);
1085             tft.setCursor(10,40);
1086             tft.print("**Tension Arterial");
1087
1088             //Ahora mandare a pantalla el texto predeterminado de inicio.
1089             //if(contador==1){
1090             //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fondo
1091             //y resalto solo el último dato a mostrar o imprimir en pantalla.
1092             tft.setTextSize(3);

```

```

1091         //tft.setTextColor(IL19341_RED);
1092         tft.setTextColor(IL19341_YELLOW);
1093         tft.setCursor(30,40);
1094         tft.print("Signos Vitales");
1095
1096         tft.setTextSize(3);
1097         //tft.setTextColor(IL19341_RED);
1098         tft.setTextColor(IL19341_GREEN);
1099         tft.setCursor(10,100);
1100         tft.print("Esperando Comando~App");
1101         //contador=0;
1102         //};
1103         //End del bloque
1104         break;
1105     }
1106 }
1107     delay(1000);
1108 }
1109 }
1110 }//Fin del bucle while principal
1111 delay(100);
1112 } //Fin del loop
1113
1114
1115 void limpioDataDesconexionSensorSpO2(){
1116     //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fondo
1117     //y resalto solo el último dato a mostrar o imprimir en pantalla.
1118     tft.setTextSize(3);
1119     //tft.setTextColor(IL19341_RED);
1120     tft.setTextColor(IL19341_BLACK);
1121
1122     tft.setCursor(30,40);
1123     tft.print("Signos Vitales");
1124
1125     tft.setTextSize(3);
1126     //tft.setTextColor(IL19341_RED);
1127     tft.setTextColor(IL19341_BLACK);
1128     tft.setCursor(10,100);
1129     tft.print("Esperando Comando~App");
1130     //End del bloque
1131     //contador=1;
1132
1133     tft.setTextSize(5);
1134     //tft.setTextColor(IL19341_RED);
1135     tft.setTextColor(IL19341_BLACK); //limpiar
1136     tft.setCursor(30,80);
1137
1138     tft.print(spo2);
1139     tft.print(" %");
1140
1141     tft.setTextSize(5);
1142     //tft.setTextColor(IL19341_RED);
1143     tft.setTextColor(IL19341_BLACK); //limpiar
1144     tft.setCursor(30,80);
1145     tft.print(spo21);
1146     tft.print(" %");
1147
1148     tft.setTextSize(5);
1149     //tft.setTextColor(IL19341_RED);
1150     tft.setTextColor(IL19341_BLACK); //limpiar
1151     tft.setCursor(30,140);
1152     tft.print(fc);

```

```

1151 tft.print(" PRbpm");
1152
1153 tft.setTextSize(5);
1154 //tft.setTextColor(IL19341_RED);
1155 tft.setTextColor(IL19341_BLACK); //limpiar
1156 tft.setCursor(30,140);
1157 tft.print(fc1);
1158 tft.print(" PRbpm");
1159
1160 tft.setTextSize(3);
1161 tft.setTextColor(IL19341_YELLOW);
1162 tft.setCursor(30,40);
1163 tft.print("<<<OXIMETRIA>>>");
1164
1165 contador=0;

1166 }
1167
1168
1169 void limpioDataDesconexionSensorTA() {
1170 //Begin Bloque:Con este bloque limpio el texto o lo oculto pintandole del mismo color del fondo
1171 //y resalto solo el último dato a mostrar o imprimir en pantalla.
1172 tft.setTextSize(3);
1173 //tft.setTextColor(IL19341_RED);
1174 tft.setTextColor(IL19341_BLACK);
1175 tft.setCursor(30,40);
1176 tft.print("Signos Vitales");
1177
1178 tft.setTextSize(3);
1179 //tft.setTextColor(IL19341_RED);
1180 tft.setTextColor(IL19341_BLACK);

1181 tft.setCursor(10,100);
1182 tft.print("Esperando Comando~App");
1183 //End del bloque
1184 //contador=1;
1185
1186 tft.setTextSize(5);
1187 //tft.setTextColor(IL19341_RED);
1188 tft.setTextColor(IL19341_BLACK); //limpiar
1189 tft.setCursor(30,80);
1190 tft.print(diastolic);
1191 tft.print(" mmHg");
1192
1193 tft.setTextSize(5);
1194 //tft.setTextColor(IL19341_RED);
1195 tft.setTextColor(IL19341_BLACK); //limpiar

1196 tft.setCursor(30,80);
1197 tft.print(diastolic1);
1198 tft.print(" mmHg");
1199
1200 tft.setTextSize(5);
1201 //tft.setTextColor(IL19341_RED);
1202 tft.setTextColor(IL19341_BLACK); //limpiar
1203 tft.setCursor(30,140);
1204 tft.print(systolic);
1205 tft.print(" mmHg");
1206
1207 tft.setTextSize(5);
1208 //tft.setTextColor(IL19341_RED);
1209 tft.setTextColor(IL19341_BLACK); //limpiar
1210 tft.setCursor(30,140);

```



```

1211 tft.print(systolic1);
1212 tft.print(" mmHg");
1213
1214 tft.setTextSize(5);
1215 //tft.setTextColor(ILI9341_RED);
1216 tft.setTextColor(ILI9341_BLACK); //limpiar
1217 tft.setCursor(30,140);
1218 tft.print(pulseMin);
1219 tft.print(" bpm");
1220
1221 tft.setTextSize(5);
1222 //tft.setTextColor(ILI9341_RED);
1223 tft.setTextColor(ILI9341_BLACK); //limpiar
1224 tft.setCursor(30,140);
1225 tft.print(pulseMin1);

1226 tft.print(" bpm");
1227
1228 tft.setTextSize(3);
1229 tft.setTextColor(ILI9341_BLACK);
1230 tft.setCursor(10,40);
1231 tft.print("<Tension Arterial>");
1232
1233 }

```

Fig. 46. Código fuente.

Partes del circuito.

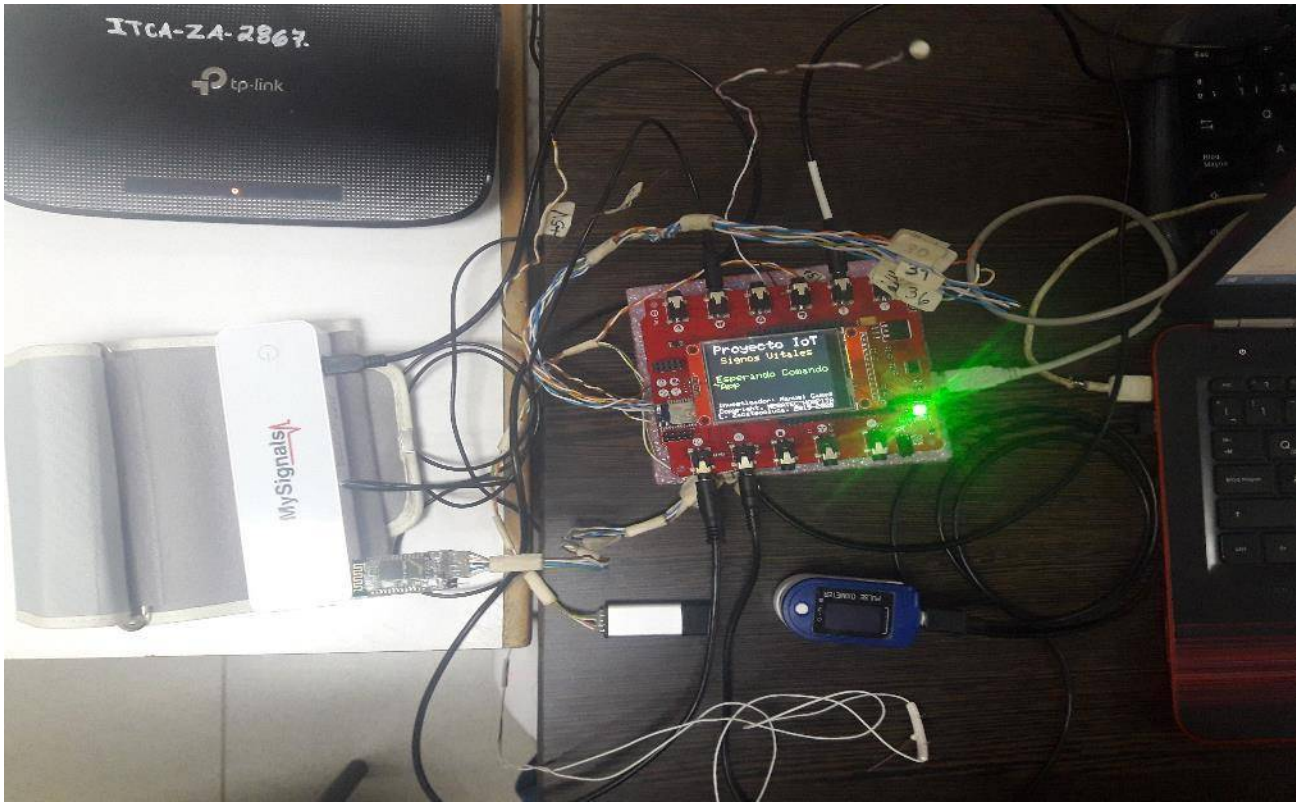


Fig. 47. Mensaje de inicio en pantalla TFT.

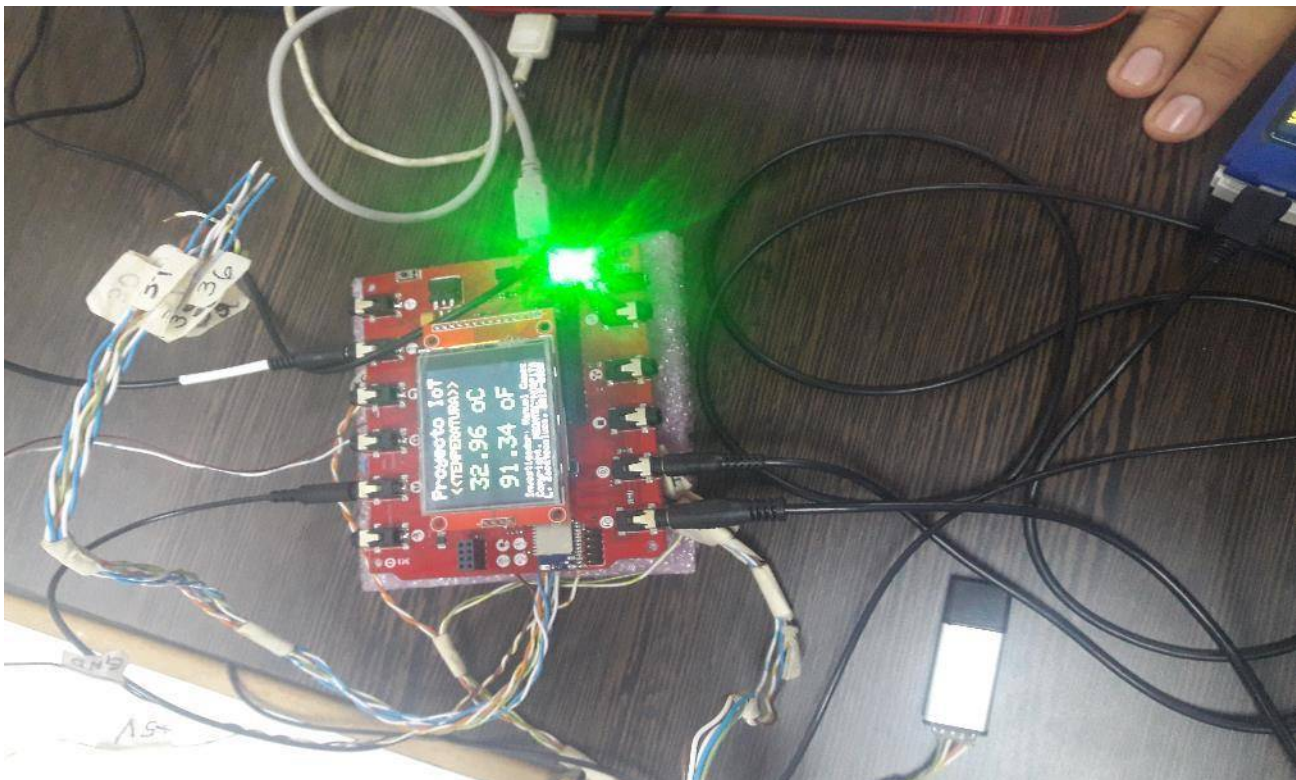


Fig. 48. Pantalla TFT, lectura temperatura corporal.

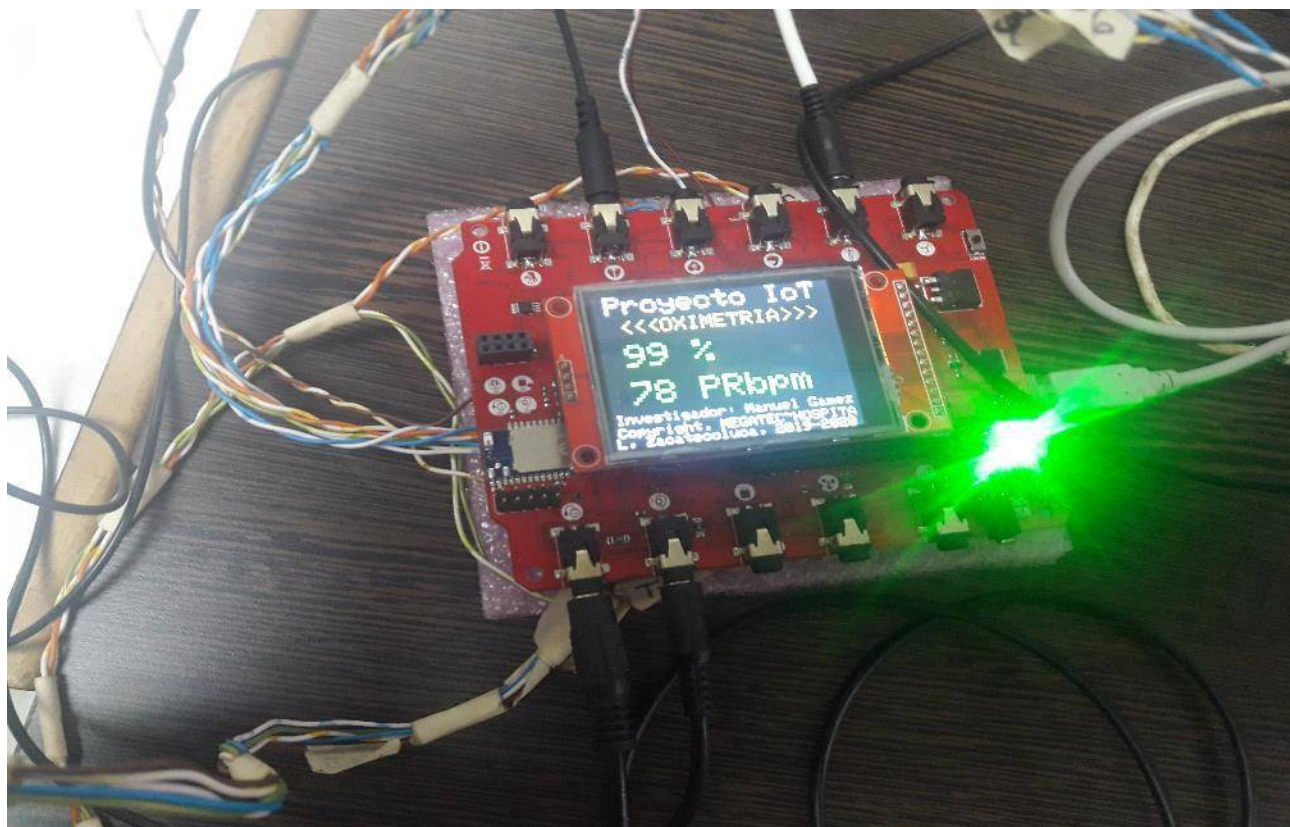


Fig. 49. Pantalla TFT, lectura oximetría.

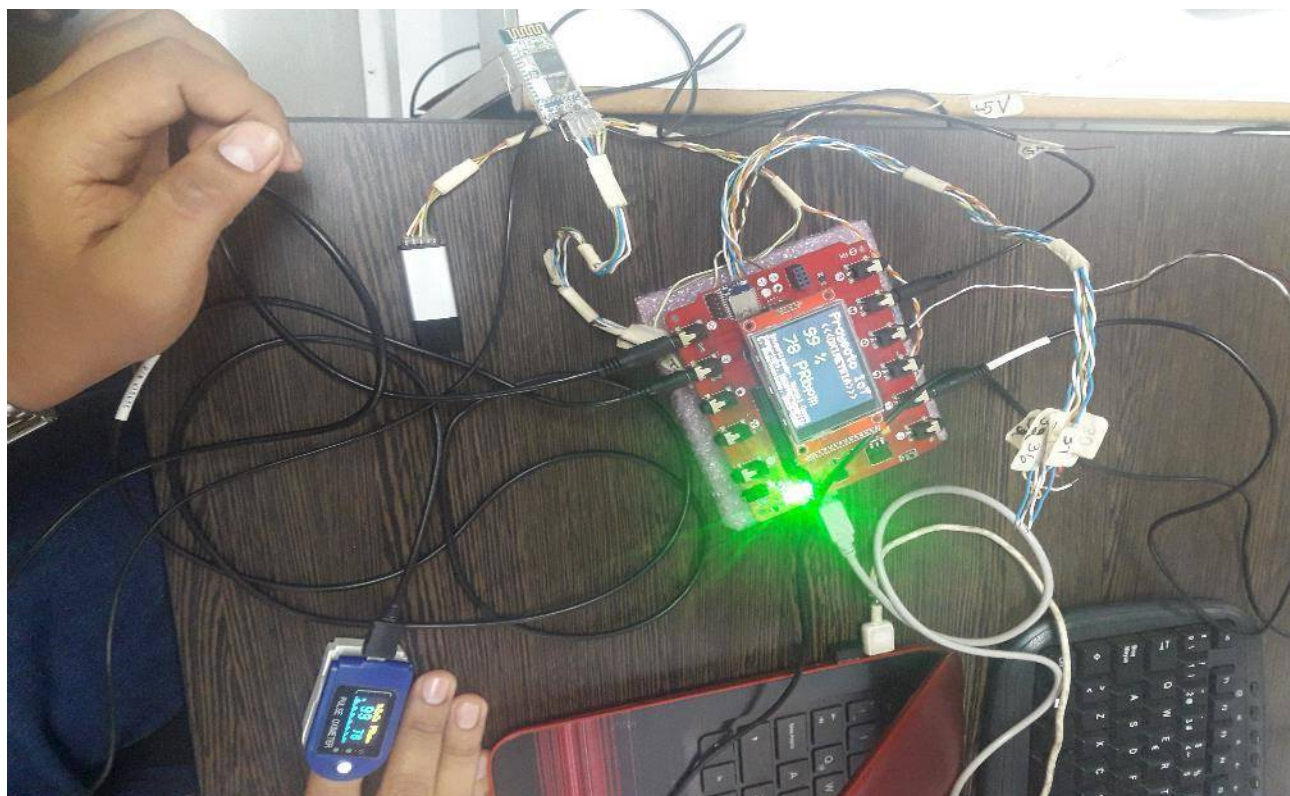


Fig. 50. Pantalla TFT, lectura oximetría (sensor).

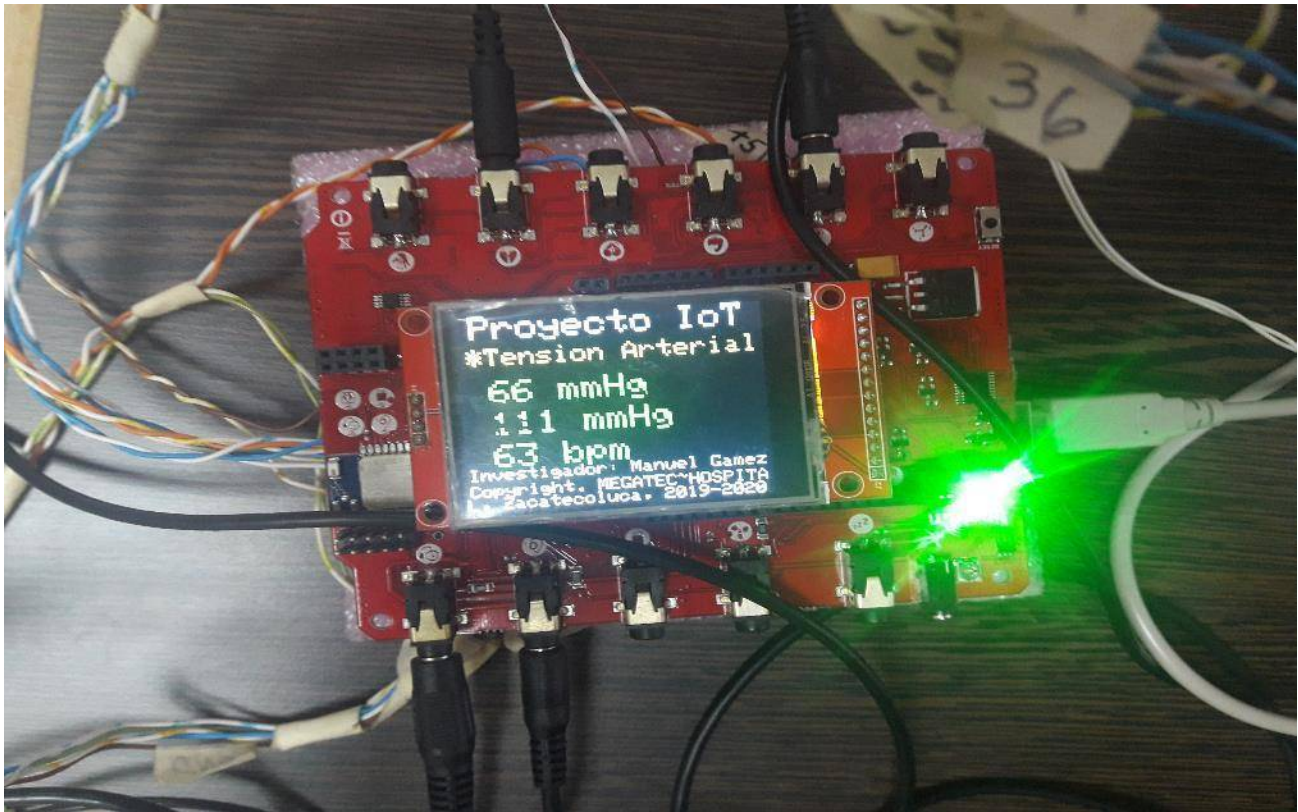


Fig. 51. Pantalla TFT, lectura de presión arterial.



Fig. 52. Paciente – Investigador.



Fig. 53. Conexión de sensores en paciente y visualización de lecturas en servidor remoto.



Fig. 54. Reunión con especialista atención sanitaria.

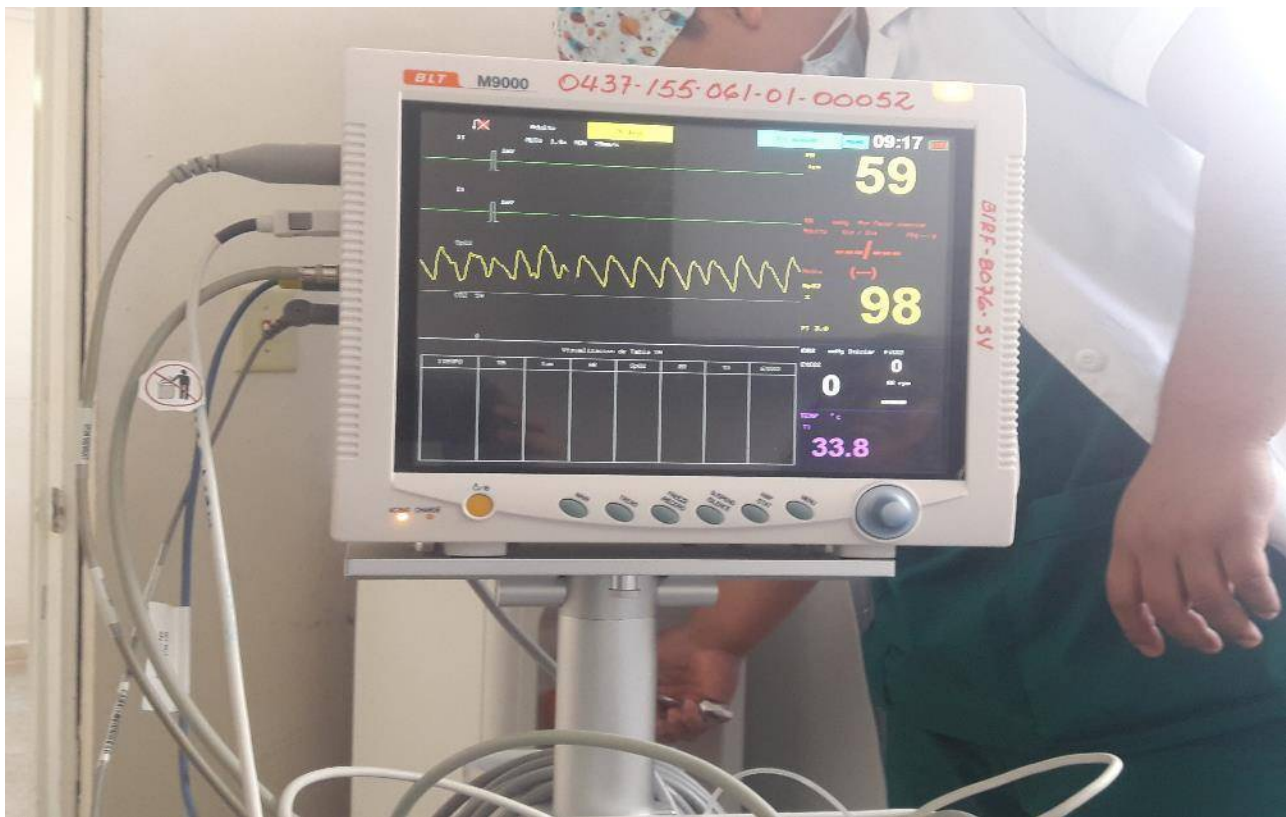


Fig. 55. Monitor especializado M9000, Utilizado para comparación de lecturas biométricas.

Lectura de Temperatura Corporal y Oximetría

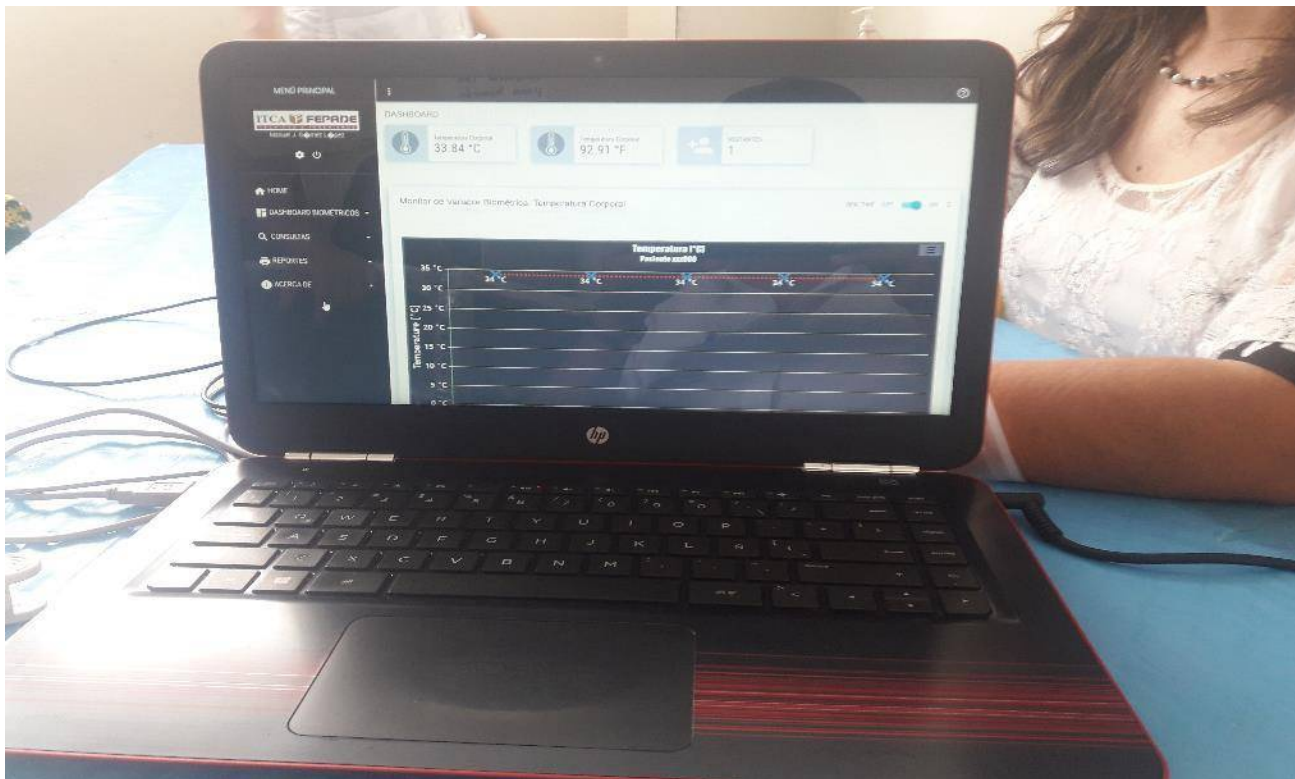


Fig. 56. Visualización de información TC, en Portal Web.



Fig. 57. Visualización de información oximetría, en Portal Web.



Fig. 58. Visualización de la información oximetría, en sensor.

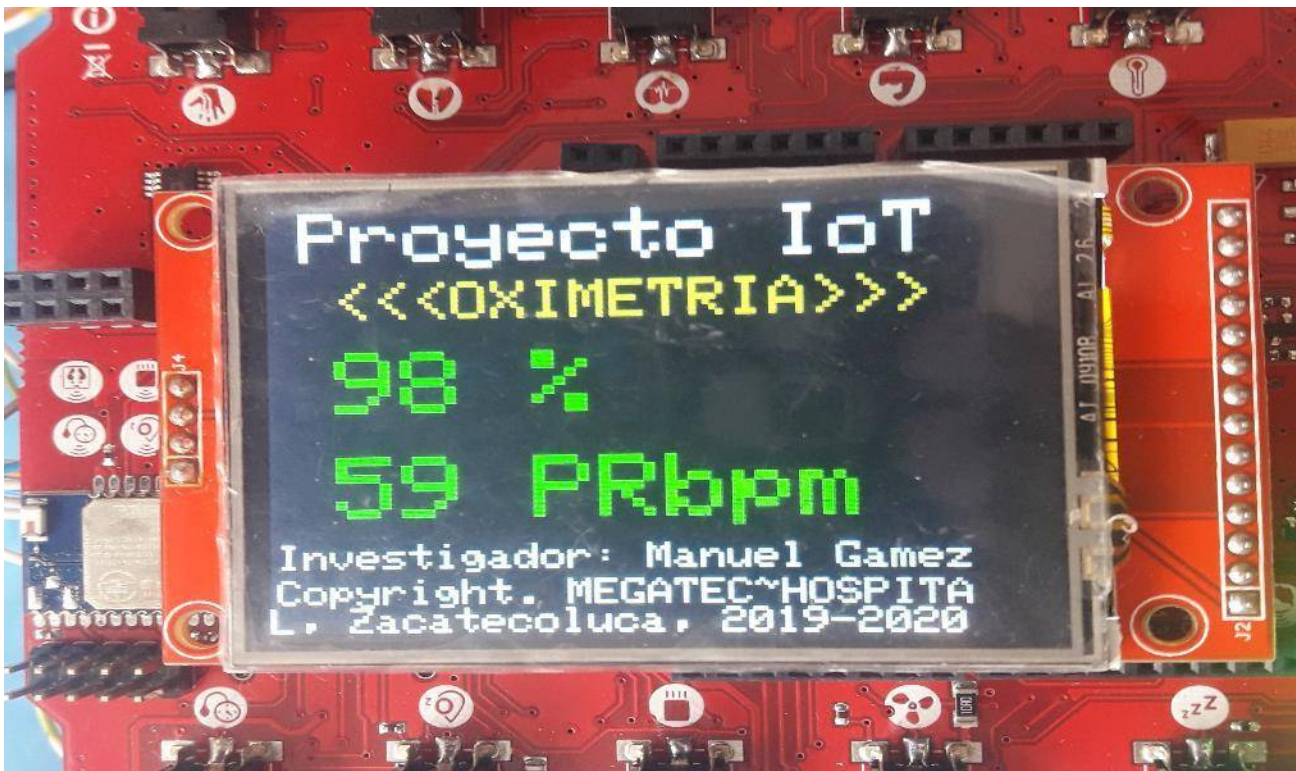


Fig. 59. Visualización de la información oximetría, en pantalla TFT.



Fig. 60. Demostración de uso de App a especialista.

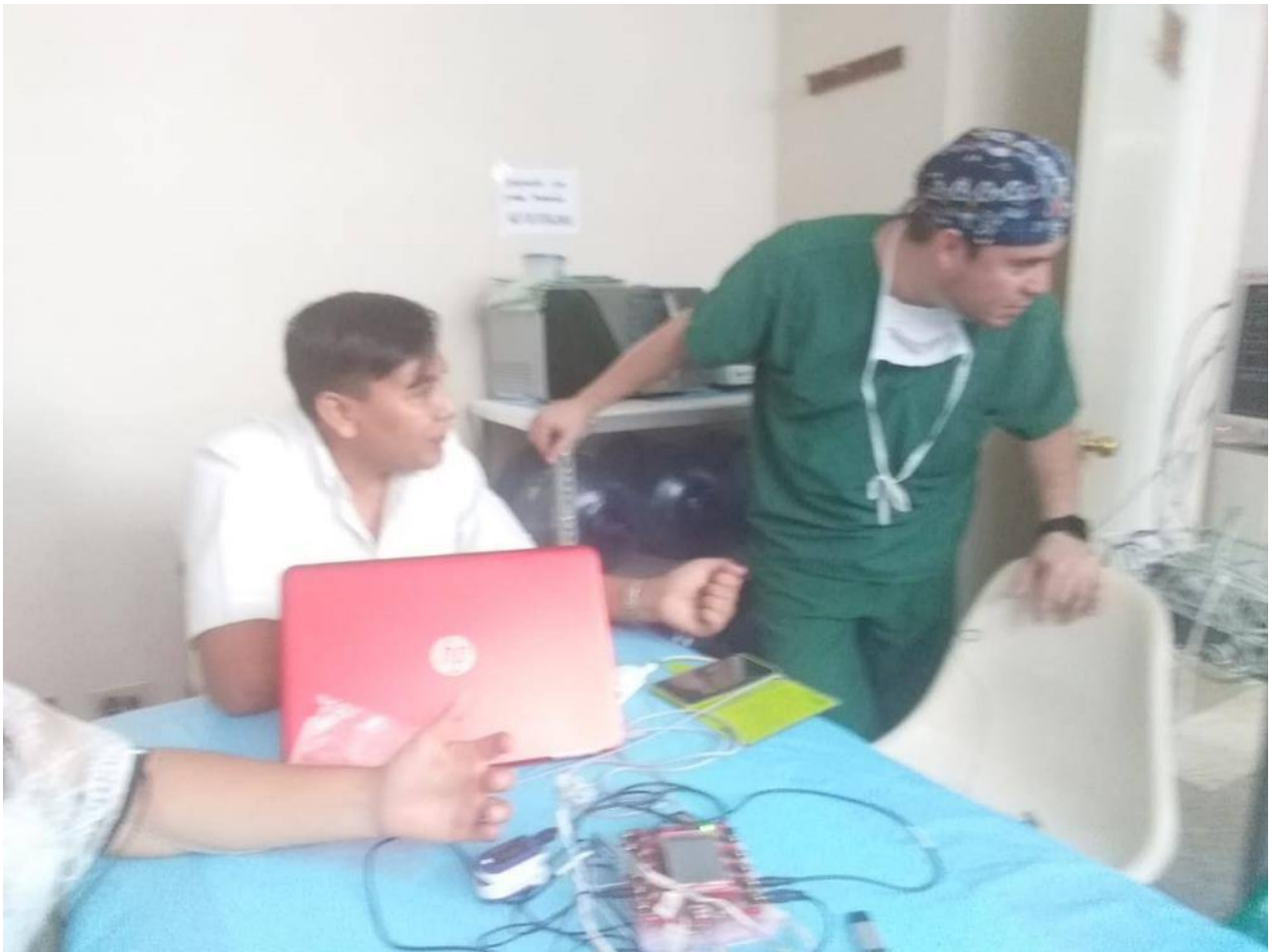


Fig. 61. Prototipo en fase de pruebas y demostración.

Tabla 10. Parámetros médicos estandarizados.

TABLAS DE SIGNOS VITALES POR EDADES

TENSION ARTERIAL				
Grupo	Edad	Rango		
		Sistólica	/	Diastólica
RN	Nacimiento – 6 semanas	70-100	/	50-68
Infante	7 semanas - 1 año	84-106	/	56-70
Lactante mayor	1 – 2 años	98-106	/	58-70
Pre-escolar	2 – 6 años	99-112	/	64-70
Escolar	6 – 13 años	104-124	/	64-86
Adolescente	13 – 16 años	118-132	/	70-82
Adulto	16 años y más	110-140	/	70-90

FRECUENCIA RESPIRATORIA		
Grupo	Edad	Ventilaciones por minuto
RN	Nacimiento – 6 semanas	40-45
Infante	7 semanas - 1 año	20-30
Lactante mayor	1 – 2 años	20-30
Pre-escolar	2 – 6 años	20-30
Escolar	6 – 13 años	12-20
Adolescente	13 – 16 años	12-20
Adulto	16 años y más	12-20

FRECUENCIA CARDIACA		
Grupo	Edad	Latidos por minuto
RN	Nacimiento – 6 semanas	120-140
Infante	7 semanas - 1 año	100-130
Lactante mayor	1 – 2 años	100-120
Pre-escolar	2 – 6 años	80-120
Escolar	6 – 13 años	80-100
Adolescente	13 – 16 años	70-80
Adulto	16 años y más	60-80

TEMPERATURA		
Grupo	Edad	Grados Centígrados
RN	Nacimiento – 6 semanas	38
Infante	7 semanas - 1 año	37.5 a 37.8
Lactante mayor	1 – 2 años	37.5 a 37.8
Pre-escolar	2 – 6 años	37.5 a 37.8
Escolar	6 – 13 años	37 a 37.5
Adolescente	13 – 16 años	37
Adulto	16 años y más	36.2 a 37.2

SEDE CENTRAL Y CENTROS REGIONALES



La Escuela Especializada en Ingeniería ITCA-FEPADE, fundada en 1969, es una institución estatal con administración privada, conformada actualmente por 5 campus: Sede Central Santa Tecla y cuatro centros regionales ubicados en Santa Ana, San Miguel, Zacatecoluca y La Unión.

1 SEDE CENTRAL SANTA TECLA

Km. 11.5 carretera a Santa Tecla, La libertad.
Tel.: (503) 2132-7400

2 CENTRO REGIONAL SANTA ANA

Final 10a. Av. Sur, Finca Procavia.
Tel.: (503) 2440-4348

3 CENTRO REGIONAL LA UNIÓN

Calle Sta. María, Col. Belén, atrás del Instituto Nacional de La Unión.
Tel.: (503) 2668-4700

4 CENTRO REGIONAL ZACATECOLUCA

Km. 64.5, desvío Hacienda El Nilo sobre autopista a Zacatecoluca.
Tel.: (503) 2334-0763 y
(503) 2334-0768

5 CENTRO REGIONAL SAN MIGUEL

Km. 140 carretera a Santa Rosa de Lima.
Tel.: (503) 2669-2298