

PROYECTO FINAL DE INGENIERÍA

SISTEMA PARA MONITOREO DE LA SALUD BASADO EN IoT LOW-COST

Figuroa Greco, Juan Luis – LU: 1107408

Ingeniería electrónica

Cibran, Juan Cruz – LU: 1099656

Ingeniería electrónica

Tutor:

Esposito, Pablo, Facultad de ingeniería - UADE

Co-Tutor:

Manuelian, Alex, Facultad de ingeniería - UADE

Diciembre 15, 2021

UADE

UNIVERSIDAD ARGENTINA DE LA EMPRESA

FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS

AGRADECIMIENTOS

En primer lugar queremos agradecer a nuestra familia, tanto hermanos como padres que siempre nos dieron su apoyo en todo momento.

A todo el personal docente de la Universidad Argentina de la Empresa (UADE) y la Universidad Simón Bolívar (USB), que sin su apoyo y conocimientos no hubiera sido posible llevar a cabo este proyecto de grado.

A nuestros compañeros de estudio y de vida durante los últimos años de carrera, que influyeron directa e indirectamente en este largo y anhelado proceso.

Muchas gracias a todos.

Juan Luis Figuroa Greco y Juan Cruz Cibran Yunes

“Un ser sin estudios, es un ser incompleto.”

— Simón Bolívar

INDICE

RESUMEN.....	7
ABSTRACT	8
1. INTRODUCCION.....	9
2. ESTADO DEL ARTE	12
2.1. Tecnología aplicada a la salud	12
2.2. Tipos de dispositivos IoT	15
2.3. Ejemplos concretos.....	16
2.4. Dispositivos de todo tipo	17
2.5. Plataformas virtuales	19
2.6. Problemática.....	20
3. ANÁLISIS DE MERCADO	21
3. METODOLOGÍA DE DESARROLLO	33
3.1. Microprocesador ESP8266 NodeMCU	33
3.2. Sensores	35
3.3. Sensor de electrocardiograma AD8232	35
3.4. Sensor de frecuencia cardiaca y pulsioximetría MAX30102.....	37
3.5. Sensor de caída MPU6050.....	39
3.6. Sensor de ubicación GPS Module (GY-NEO6MV2).....	40
3.7. Sensor de temperatura infrarrojo MLX90614.....	41
3.8. Servidor.....	42
3.9. Almacenamiento de Datos	43
3.10. UBIDOTS.....	43
3.11. IFTTT	44
4. PRUEBAS REALIZADAS.....	46
4.1. Funcionamiento del ECG AD8232.....	46
4.2. Funcionamiento del MLX90614	52
4.3. Funcionamiento del MAX30102.....	56
4.4. Funcionamiento del GY-GPS6MV2	59
4.5. Funcionamiento del MPU6050	62

5. ANÁLISIS COMERCIAL.....	65
5.1. Análisis de precios	65
5.2. Análisis de estrategias (FODA estratégico).....	66
6. DISCUSIÓN	67
GLOSARIO.....	70
BIBLIOGRAFÍA.....	71
ANEXO A.....	73
ANEXO B	79
ANEXO C	84
ANEXO D.....	85
ANEXO E	88

INDICE DE FIGURAS

Figura 1: Mapa mental de sensores y salida	10
Figura 2: Glucómetro Enlite, MiniMed 640G Sistema Integrado.....	12
Figura 3: Valor estimado por continente	13
Figura 4: Pastillas inteligentes.....	17
Figura 5: Audiómetro digital Audixi 10	17
Figura 6: Respiro™ de Amiko	18
Figura 7: MoMe Kardia	19
Figura 8: Intervalo de edades	21
Figura 9: Lugar de residencia de los encuestados	22
Figura 10: Tipo de servicio, público o privado	23
Figura 11: Porcentaje de médicos o personal de la salud	24
Figura 12: Porcentaje de lugar de trabajo, público o privado	25
Figura 13: Conoces las siglas “IoT”	26
Figura 14: Trabajas con algún equipo IoMT.....	27
Figura 15: Parámetros medidos al paciente	28

Figura 16: Conoces las siglas IoT	30
Figura 17: Te gustaría usar esta tecnología.....	31
Figura 18: Dejarías que un familiar en estado crítico sea observado de esta manera	32
Figura 19: ESP8266 NodeMCU.....	34
Figura 20: Diagrama de pinout de NodeMCU	34
Figura 21: ECG Sensor AD8232.....	35
Figura 22: ECG Gráfica y explicación	36
Figura 23: Sensor MAX30102	38
Figura 24: Sensor MPU6050.....	39
Figura 25: Sensor GY-NEO6MV2.....	40
Figura 26: Sensor MLX90614.....	41
Figura 27: Mapa mental de la página web IFTTT.....	44
Figura 28: Esquema de conexión AD8232	46
Figura 29: Triángulo de Einthoven.....	47
Figura 30: Ubicación de almohadillas y salida de ECG	48
Figura 31: Gráfica prueba ECG.....	49
Figura 32: Electrocardiograma en perfecto funcionamiento.....	49
Figura 33: Salida del monitor serie para el envío de datos del ECG.....	50
Figura 34: Ubicación elegida para almohadillas ECG	51
Figura 35: Electrocardiograma del paciente en la pantalla de Ubidots	52
Figura 36: Dashboard del electrocardiograma del paciente.....	52
Figura 37: Esquema de conexión MLX90614-NodeMCU-OLED	53
Figura 38: Monitor Serie MLX90614.....	54
Figura 39: Temperatura medida y observada mediante Ubidots.....	55
Figura 40: Dashboard del Ubidots con ECG y temperatura corporal.....	55
Figura 41: Dashboard completo del Ubidots con ECG y temperatura corporal	56
Figura 42: Esquema conexión MAX30102	56
Figura 43: Monitor serie con la salida de oxígeno en la sangre y pulso del paciente	57
Figura 44: Valores de Oxígeno en la sangre y pulso del paciente	58

Figura 45: Dashboard completo de Ubidots	59
Figura 46: Esquema conexión GY-GPS6MV2 con NodeMCU.....	60
Figura 47: Salida puerto serie al momento de enviar la ubicación	60
Figura 48: Interfaz de módulo GPS para la ubicación.....	61
Figura 49: Ubicación del paciente prueba	62
Figura 50: Esquema de conexión MPU6050 con NodeMCU	63
Figura 51: Salida del puerto serie al momento de detectar la caída del paciente.....	64
Figura 52: Notificación de la caída del paciente	64

INDICE DE TABLAS

Tabla I: Lista de materiales y costo del proyecto	65
Tabla II: FODA estratégico.....	66

RESUMEN

La monitorización continua de los parámetros fisiológicos es fundamental para los pacientes ancianos y enfermos, mediante la observación y el registro continuo de los parámetros se valora el estado actual del paciente, su evolución y la repercusión de la terapéutica en su hemodinámica, lo cual le dará la oportunidad al médico para interpretar, detectar y evaluar los problemas y actuar de forma eficaz.

Existe la necesidad de un sistema de monitorización eficiente de la salud del paciente basado en Cloud, cuando el paciente no se encuentra en el centro de salud. Este sistema les permitirá a los médicos controlar los parámetros fisiológicos normales para cada paciente de manera online y poder tomar las medidas necesarias en caso de emergencia.

En este proyecto se implementa un sistema de monitoreo de parámetros fisiológicos basado NodeMCU y la plataforma Ubidots. Varios parámetros básicos como ECG, la presión arterial, la glucosa, la temperatura corporal entre otros, serán censados y procesados para poder visualizarlos ya sea en la pantalla LCD del aparato como en el Cloud al que tiene ingreso el médico o personal de la salud a distancia.

ABSTRACT

The continuous monitoring of the physiological parameters is essential to obtain a continuous record of the parameters and the assessment of the current state of the patient, especially in the elderly and sick, the evolution of the same and the impact of the therapy on his hemodynamics; Giving the specialist the opportunity to detect, evaluate and interpret any irregularity and act effectively.

There is a need to create an efficient monitoring system of the patient's health status through the cloud, without the need for the presence of the same. This system will allow medical personnel to control the physiological parameters of each patient online and to act in times of emergencies. In this project, a physiological parameters monitoring system based NodeMCU and the Ubidots platform will be implemented.

A series of basic parameters such as ECG, blood pressure, glucose, body temperature among others, will be sensed and processed to be able to view through the LCD screen of the device as in the cloud, to which the medical team will have access at a distance.

1. INTRODUCCION

En la nueva era de la comunicación y la tecnología, internet se ha entrelazado con la mayoría de los aspectos de nuestra vida cotidiana a través del crecimiento desmedido de los dispositivos electrónicos. Esto facilitó que se abrieran distintos caminos para el uso de servicios en la nube (Cloud).

Como en la mayoría de las distintas industrias, la utilización de las tecnologías de la información (IT) y las nuevas tecnologías se están materializando dentro del sector de la salud. A medida que el envejecimiento de la población mundial está creciendo y las enfermedades crónicas aumentan rápidamente en todo el mundo, esto le genera una carga al sistema de salud y ejerce más presión sobre los proveedores de atención médica y los médicos para que encuentren soluciones.

Uno de los conceptos que surgen con las Tecnologías de la Información (TIC) y que está creciendo desmedidamente por sus posibilidades en solucionar los problemas anteriormente mencionados dentro de la salud es el Internet de las cosas (IoT). Con el aumento de los dispositivos conectados, combinado con el avance de los sistemas para capturar y transmitir los datos, esto ha facilitado el surgimiento de Internet de las cosas aplicado a la medicina (IoMT). Dado que la tecnología de detección inteligente recopila información en tiempo real, esto permite que se pueda investigar y pronosticar elementos médicos.

IoMT es una infraestructura conectada de dispositivos médicos, aplicaciones de software y sistemas y servicios de salud. La cantidad de dispositivos conectados a la red y que conforman el Internet de las cosas (IoT), continúa creciendo a un ritmo desmedido. De acuerdo con las previsiones del IDC en 2025 habrá 42.600 millones de dispositivos conectados a IoT que generarán 79,4 Zettabytes de datos.

Con el surgimiento de nuevas aplicaciones de tecnologías dentro de la salud, esto también desafía los modelos tradicionales al proporcionar nuevos modelos para la creación de valor dentro de los ecosistemas de salud. Dado que muchos actores interactúan dentro de estos ecosistemas, sería buena idea realizar más investigaciones que examinen qué factores de contexto influyen en la implementación de nuevas tecnologías, con el resultado de aliviar el modelo de atención médica tradicional.

Nuestro sistema IoMT, está basado en el microcontrolador NodeMCU, el cual controla una serie de sensores descritos más adelante y son los encargados de obtener toda la información del paciente ya sea analógica o digitalmente, estos parámetros podrán ser procesados y enviados a la nube (Cloud) gracias al placa, el cual es una plataforma IoT de código abierto que incluye el firmware que se ejecuta en el SoC (Sistema en chip) Wi-Fi ESP8266 de “Espressif Systems” y el hardware que se basa en el módulo ESP-12. Con la ayuda de todos estos elementos, logramos ya tener en “internet” los datos censados en el paciente, y para poder visualizarlos utilizamos Ubidots que es una plataforma de IoT, que permite recoger y almacenar datos de sensores en la nube y desarrollar aplicaciones IoT. Ubidots también ofrece aplicaciones que permiten analizar y visualizar tus datos.

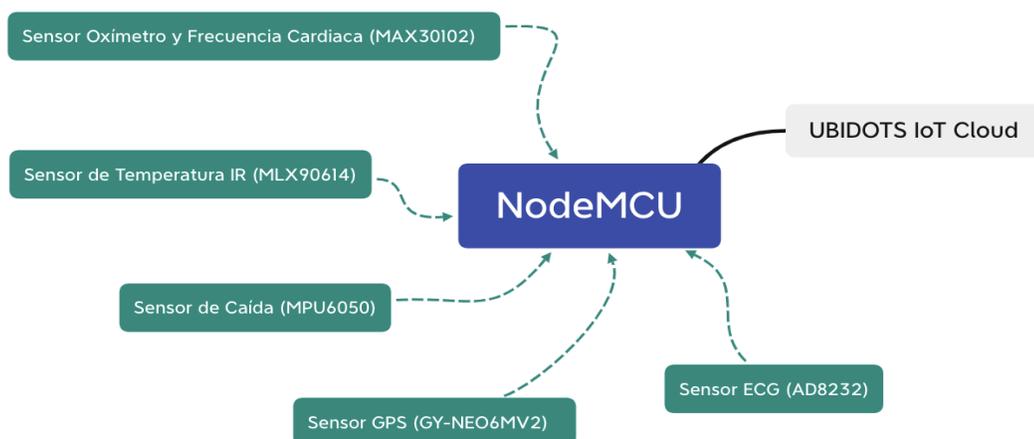


Figura 1: Mapa mental de sensores y salida

El proyecto estará basado en tres grandes etapas, la primera será evaluar técnica y económicamente tanto los sensores, microcontroladores y plataformas IoT que serán utilizadas y el por qué para llevar a cabo proyecto, luego como segunda etapa, se mostrará el proceso de desarrollo en el cual se integrarán todos los sensores y su uso correcto al microprocesador elegido, así como la codificación del NodeMCU, por último, se dará muestra de todos los sensores en la plataforma seleccionada que recoge los datos y la mejor forma de visualizarlos, en esta parte será explicado brevemente que detallan estos resultados, que significan para el médico o personal elegido y el porqué de medir estos parámetros fisiológicos en las personas, así como el correcto uso de estos.

Para el desarrollo del dispositivo con las características mencionadas anteriormente, se estructurará el informe de la siguiente manera:

- Estado del Arte: Se hará en primer lugar un estudio de los antecedentes. Esto es, una revisión de las soluciones principales que se comercializan hoy en día y una descripción de sus características principales.
- Contexto: Se desarrollarán teóricamente conceptos tecnológicos relacionados a la materia de estudio, necesarios para entender el desarrollo de las secciones siguientes.
- Análisis de mercado: Se realizará una encuesta a la sociedad con el objetivo de entender la presencia del IoT en Argentina, cuanto entendimiento hay sobre el tema y las preferencias de los usuarios.
- Metodología de desarrollo: Se explicará el proceso de desarrollo del prototipo principal y sus periféricos tanto desde el punto de vista de software como de hardware.
- Análisis comercial: Se realizará un estudio sobre el precio de la solución desarrollada en el presente proyecto.
- Discusión: Se elaborarán comentarios conclusivos acerca de la realización de este trabajo, analizando los resultados de las pruebas y limitaciones del prototipo.

Cabe aclarar que, todos los precios de dispositivos y componentes reflejados en el presente informe, fueron tomados en el mes de agosto de 2020.

2. ESTADO DEL ARTE

2.1. Tecnología aplicada a la salud

La tecnología avanza a pasos agigantados, en todos los sectores, la salud no es la excepción, muchos de sus procedimientos, dispositivos y medicamentos han obtenido mejoras muy significativas. La realidad virtual, el Big Data, la AI (inteligencia artificial) y el internet de las cosas están suponiendo una gran mejora para el sector.

La tecnología aplicada a la salud tiene una clara representación en el uso de dispositivos médicos capaces de monitorizar pacientes y enviar datos de los mismos en tiempo real para poder evaluar su estado de salud, e intervenir incluso antes de que se produzca un problema, algo que transformará la forma en que se organiza y se concibe la medicina.. Lo cierto es que la tecnología es parte fundamental de nuestros sistemas sanitarios y una de las claves principales de su eficiencia.



Figura 2: Glucómetro Enlite, MiniMed 640G Sistema Integrado

Los principales avances en tecnología están impulsando la innovación en tecnología médica, lo que lleva al desarrollo de un número cada vez mayor de dispositivos médicos conectados que pueden generar, recopilar, analizar y transmitir datos. Los datos, junto con los dispositivos en sí, están creando el IoMT, una infraestructura conectada de dispositivos médicos, aplicaciones de software y sistemas y servicios de salud. Se estima que esta infraestructura conectada de

dispositivos médicos tendrá un valor estimado de 158.1 mil millones de dólares en 2022. Algunos ejemplos de dispositivos IoMT los podemos encontrar en inhaladores inteligentes para mejora de tratamientos respiratorios, medidores de glucosa en sangre que envían los datos directamente al profesional sanitario o sistemas de administración de medicamentos (para enviar alertas a los pacientes sobre sus tratamientos), entre otros.



Figura 3: Valor estimado por continente

Las empresas de tecnología médica fabrican una amplia variedad de dispositivos médicos (se cifra en más de medio millón), como marcapasos, sistemas de monitoreo domésticos o monitores de glucosa en sangre, entre muchos otros.

Los principales beneficios que aporta IoMT son:

- Mejorar los diagnósticos: en un contexto como el actual, contar con tests dotados de conexión a internet permitiría realizar pruebas de contagio del Covid-19 a distancia. Agilizando, así, el proceso de diagnóstico y limitando los contagios. Se reducen los errores de diagnóstico y aplicación de tratamientos. Aportan información sobre el estado de salud del paciente en tiempo real
- Monitorizar las enfermedades: algo factible tanto para enfermedades temporales como crónicas. Puede resultar determinante para hacer seguimientos de pacientes cuyas constantes deben ser medidas con frecuencia o de aquellos que se encuentran en sus

domicilios. Permiten realizar un seguimiento personalizado del paciente y corregir tratamientos de forma inmediata.

- Hacer un seguimiento de los datos: los dispositivos conectados pueden extraer datos e información de forma continuada y compartirlas en tiempo real. Permiten al personal sanitario hacer un seguimiento más exhaustivo y completo y personalizar los tratamientos. Además, contar con información en tiempo real da la opción de adaptarse a los cambios de forma más inmediata.
- Registrar la información: las técnicas de big data permiten transformar grandes cantidades de datos en información clara. Una pandemia como la causada por este nuevo coronavirus, por ejemplo, genera gran cantidad de datos alrededor de todo el mundo. Ponerlos en orden y analizarlos de forma estructurada supone una oportunidad única para entender cómo ha evolucionado la enfermedad y medir sus consecuencias.

Estas mejoras no solo se verán aplicadas a personas, sino también tras los sanitarios transformando todo lo que hay detrás (la gestión de clínicas, hospitales y otros centros de salud).

Algunos ejemplos de estas mejoras podrían ser:

- Control del stock y gestión del material: gracias a tecnología como las etiquetas RFID, puede simplificarse en gran medida el trabajo de inventario y la gestión del material.
- Seguimiento de los pacientes: wearables y otros dispositivos permiten ubicar a los pacientes ingresados en los hospitales y compartir información sobre su historial entre el personal sanitario.
- Control y organización de los trabajadores: los dispositivos IoT permiten la optimización del flujo de trabajo de los hospitales y los centros de salud, al igual que ocurre en otros sectores.
- Mejora la experiencia del paciente en el centro sanitario: con citas personalizadas y mejor optimización del espacio de atención. Reducen el tiempo de atención médica (reducir las listas de espera de centros de salud y hospitales, o disminuir el tiempo de espera para cirugías, por ejemplo).

2.2. Tipos de dispositivos IoMT

- **Telemedicina:** La telemedicina se encarga de prestar servicios médicos y sanitarios de forma remota. De esta forma es posible proporcionar asistencia sanitaria en zonas poco accesibles, desfavorecidas económicamente o en lugares donde no se disponga de tecnología ni facultativos cualificados en un área concreta de medicina. Con IoMT se puede monitorizar a los pacientes a distancia sin necesidad de que se desplacen hacia los centros médicos para su control. Estos dispositivos permiten que la telemedicina sea interactiva, fomentando que pacientes y médicos puedan comunicarse e interactuar en tiempo real.
- **Wearables:** Los wearables son dispositivos médicos que los pacientes pueden llevar encima para medir sus datos vitales. Toda esta información es enviada a la nube por estos dispositivos IoMT y así los médicos y profesionales sanitarios tienen acceso en tiempo real a los mismos (sin necesidad de que el paciente o médico se encuentre en un centro de salud y hospital).
- **Drones:** Los drones se utilizan en el sector sanitario para acelerar la entrega de suministros y productos médicos como medicamentos, muestras (sangre, orina, o piel, por ejemplo), y otros suministros médicos. Estos dispositivos dotados de comunicación son capaces de informar sobre dónde se encuentran y el tiempo que tardan en hacer la entrega de los suministros. El tiempo en medicina es fundamental para poder realizar diagnósticos a tiempo y poder aplicar los tratamientos apropiados lo antes posible. Con el uso de IoMT como los drones, se acelera todo el proceso de entrega de suministros médicos, haciendo que todos los procesos relacionados se realicen de forma más ágil.
- **Realidad aumentada:** La realidad aumentada combina elementos de realidad virtual con entornos reales. Su uso en medicina ayuda a mejorar la seguridad del paciente y permite realizar operaciones importantes. Se utiliza de forma efectiva en el diagnóstico al poder superponer registros y signos vitales de pacientes en el proceso de evaluación del mismo. Otro aspecto donde los dispositivos IoMT de realidad aumentada tienen una gran aplicación en medicina es en el área de la formación. Permiten realizar simulaciones para capacitar a los médicos y poder educar a los pacientes con el objetivo de mejorar la aplicación y

efectividad de sus tratamientos. La realidad virtual y aumentada también se aplica en otras áreas de la medicina como en la rehabilitación y la creación de entornos para reducir el nivel de estrés y distraer del dolor y las molestias a los pacientes.

- **Inteligencia Artificial y Big Data:** Los IoT generan una gran cantidad de información sobre los pacientes que es transmitida por medio de internet, comunicándose entre dispositivos, con sistemas informáticos, o con los propios profesionales sanitarios. Esta comunicación genera una inmensa cantidad de datos que pueden ser procesados, organizados (Big Data) de manera inteligente (mediante algoritmos de inteligencia artificial) para convertir toda esa información en conocimiento que genere valor para la salud. Algunos ejemplos de los beneficios que proporciona el análisis inteligente de todos los datos que generan los dispositivos IoT lo encontramos en la detección precoz de enfermedades (gracias a detectar a tiempo sus síntomas), la aceleración a la hora de realizar diagnósticos y proporcionar un tratamiento adecuado, o la identificación de factores de riesgo de suicidio, abuso de alcohol o consumo de drogas (medicina preventiva).

2.3. Ejemplos concretos

Las “pastillas inteligentes”, el caso de Ability MyCite se conoce como el de la “primera píldora inteligente”. Desarrollada por Otsuka Pharmaceutical Co. y Ltd. y Proteus Digital Health, se trata de una medicación con aripiprazol, pensada para pacientes con problemas de esquizofrenia, trastorno bipolar y como un complemento para antidepresivos.

La pastilla contiene un sensor digital. Cuando se traga, desde el estómago envía una señal a una aplicación de teléfono, que indica a qué hora se tomó la píldora. Este tipo de dispositivos ayudan a controlar que los pacientes sigan la medicación pautada. Ability MyCite fue aprobada por la Administración de Drogas y Alimentos del Gobierno de los Estados Unidos en noviembre pasado, lo que abre la puerta a que otras empresas empiecen a desarrollar más medicación con sensores.



Figura 4: Pastillas inteligentes

En una línea muy parecida funciona el frasco de pastillas creado por AdhereTech. En este caso, es el bote el que lleva un sensor que avisa de cuándo se abre y envía alertas al paciente cuando se olvida de tomar la dosis conveniente.

2.4. Dispositivos de todo tipo

Kiversal es una empresa que ha desarrollado un audiómetro digital. Su nombre es Audixi 10 y su objetivo es ser más cómodo y manejable. Su sistema de calibración y mantenimiento le permite estar conectado permanentemente, por lo que se simplifica el proceso de suministro de datos de paciente a doctor. De esta forma, se agiliza el análisis de datos.



Figura 5: Audiómetro digital Audixi 10

Respiro™ es un producto que la empresa Amiko vende como “el futuro del cuidado respiratorio”. Es un sistema de inhaladores inteligentes programado para analizar y mejorar las terapias respiratorias. Mide el uso de medicamentos y sus resultados en la salud pulmonar de cada individuo, por lo que ofrece atención personalizada.



Figura 6: Respiro™ de Amiko

Uno de los usos más destacados del IoT de la salud se vincula a las enfermedades cardíacas. MoMe Kardía es un dispositivo que monitoriza el ritmo cardíaco y ofrece un seguimiento continuo. La empresa InfoBionic ha creado un monitor ergonómico y sencillo para que los pacientes puedan manejarlo por sí mismos.

Envía los datos a la consulta del especialista y evita que el usuario tenga que hacer constantes visitas a su doctor. Transmite los datos en tiempo real por lo que cualquier paciente estará constantemente vigilado.



Figura 7: MoMe Kardia

2.5. Plataformas virtuales

Mención especial merecen las plataformas IoT de la salud, que son el software que conecta los dispositivos, las redes de datos y las aplicaciones. Son sistemas de gestión y seguimiento. Funcionan como la esencia de la comunicación entre paciente y médico pues almacenan, administran y envían los datos de cada historial.

Y, además, incluyen herramientas de análisis y diagnóstico claves para el seguimiento de un tratamiento. Gracias a la IA y a los datos cruzados consiguen tratamientos cada vez más eficaces.

Según datos de IDC, en 2020, el uso de plataformas médicas virtuales se duplicará en todo el mundo. Y, en los próximos años, el 50% de las visitas médicas se harán a través de ellas.

Algunos ejemplos de plataformas digitales son eVisit, en la que los médicos visitan a sus pacientes, realizan diagnósticos y recetan medicinas de manera virtual. O IDOC3, una plataforma 24/7 con más de 400 médicos que atienden a pacientes de países de habla hispana como México, Argentina y Chile. Lo hacen de manera gratuita y pueden dar servicios a personas que habitan en lugares remotos, para los que visitar un médico es una misión casi imposible.

2.6. Problemática

Al enumerar los distintos productos que se encuentran en el mercado y sus funciones, nos damos cuenta que son pocas las opciones que involucran el IoT y la medicina en el mercado actual, si bien es una tecnología que ha explotado infinitamente en los últimos 3 años no ha llegado al día a día de las personas o establecimientos médicos, más aún, ver el problema que evidenció la pandemia del COVID-19 al abarrotar los hospitales y clínicas, además de la enfermedad generar un problema ocupacional de los establecimientos públicos y privados para el cuidado de la salud humana, lo cual nos dio la visión de llevar a cabo este proyecto, dirigiéndonos directamente a nuevas tecnologías y aplicaciones necesarias.

Actualmente no se cuenta con una gran cantidad de sensores que ayuden al equipo a mejorar sus funciones, así como tampoco son de muy buena calidad y precisión, lo que lo hace aún inviable aplicarlo hoy en día pues para poder atender personas y brindarle el mejor equipo que la medicina vea aceptable se necesita tener mejores sensores, que serían mucho caros y aun no se han desarrollado. Sin embargo, es un buen prototipo que deja asentada una base para la mejora de equipos y un camino viable para la solución de problemas y aplicación de nuevas tecnologías para la medicina y sistema de la salud actual.

Se realizó una encuesta para saber cómo sería visto en el mercado el prototipo trabajado, tanto como para el médico o personal de la salud como para el usuario en casa, esta encuesta nos ayudará a entender las necesidades del mercado o como sería visto por el paciente el uso de nuevas tecnologías para la medicina, esta encuesta está en momento de recolección de datos y respuestas, esperando el estimado más o menos de unos 250 a 300 respuestas para tener una buena cantidad de datos a analizar.

3. ANÁLISIS DE MERCADO

Con el fin de conocer un poco más cómo se vería el producto tanto del lado de los médicos acerca de sus funcionalidades y modo de operar, como la de los usuarios que estarían sometidos a una nueva metodología para el seguimiento de su salud se realizó un análisis a través de una encuesta que se divide dependiendo de si formas parte del personal de la salud o no, para luego separar la información en dos grandes áreas de estudio.

La encuesta fue realizada de manera online a través de Google Forms, en la cual se obtuvieron un total de 137 respuestas. De los cuales 65 pertenecían al grupo de personal de salud o médico, mientras que el resto no.

Primero, se realizaron 3 preguntas al encuestado para saber su perfil y poder tener una mayor base en cuanto a sus respuestas y el por qué.

Obteniendo los siguientes resultados:

Edad

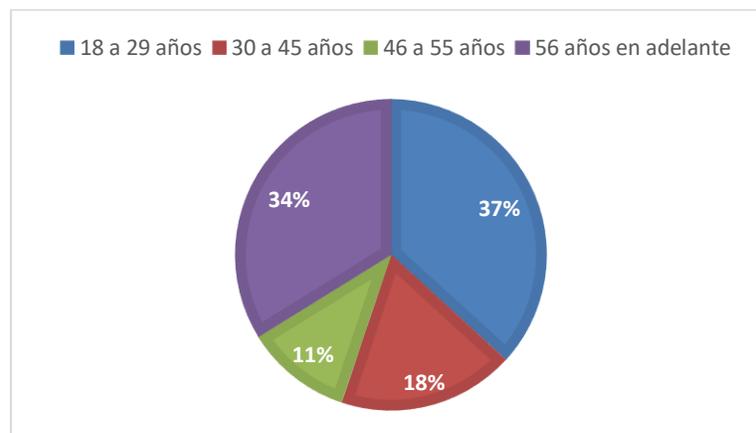


Figura 8: Intervalo de edades

Se puede notar que en su mayoría tenemos personas jóvenes menores de 30 años, y personas mayores de 56 años, que a su vez al leer las respuestas nos dimos cuenta que en su mayoría las personas mayores a 56 años pertenecían al grupo del personal de la salud mientras que los jóvenes no.

País de residencia

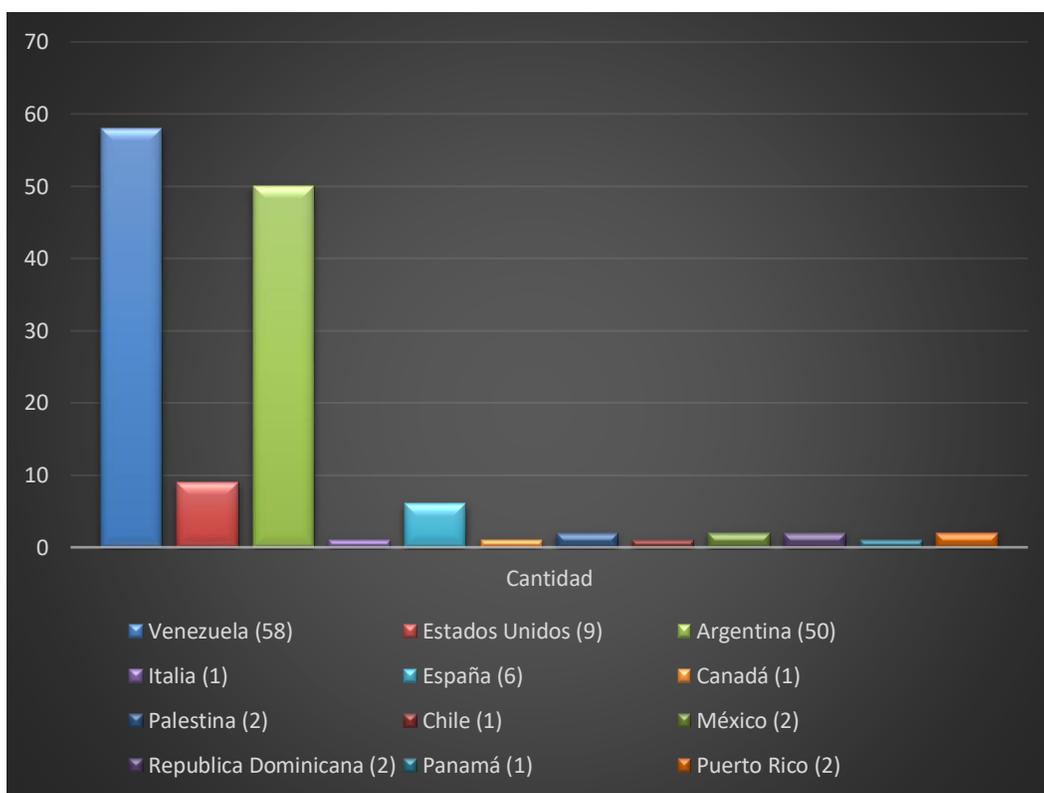


Figura 9: Lugar de residencia de los encuestados

Pudimos obtener la respuesta de las encuestas de diferentes países, lo que nos ayudará también a centrar nuestros objetivos dependiendo del nivel de desarrollo que tengan estos ya que pueden ser clasificados en “Primero, Segundo o Tercer mundo” según su nivel de desarrollo o ingresos que va ligado directamente a la calidad de los servicios médicos como de la tecnología.

¿Al momento de ir al médico, a qué tipo de medicina recurre?

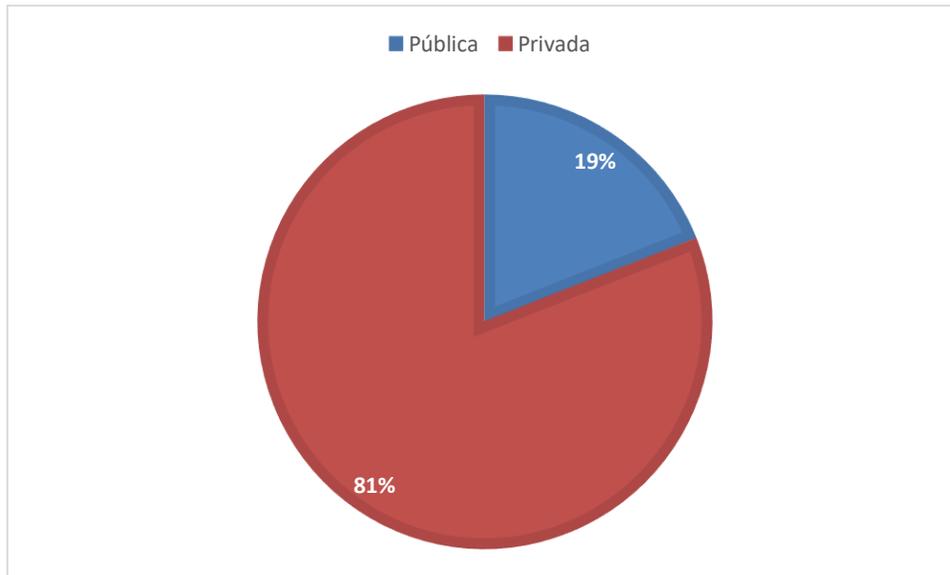


Figura 10: Tipo de servicio, público o privado

Es fácil observar, que la mayoría de nuestros encuestados suelen concurrir a establecimientos privados que por lo general suelen tener las últimas tecnologías médicas o las más avanzadas con respecto a los establecimientos públicos o del gobierno.

A partir de las siguientes preguntas, la encuesta se divide en dos, si perteneces al personal de la salud o no para luego concluir con unas preguntas generales.

¿Formas parte del personal de la salud?

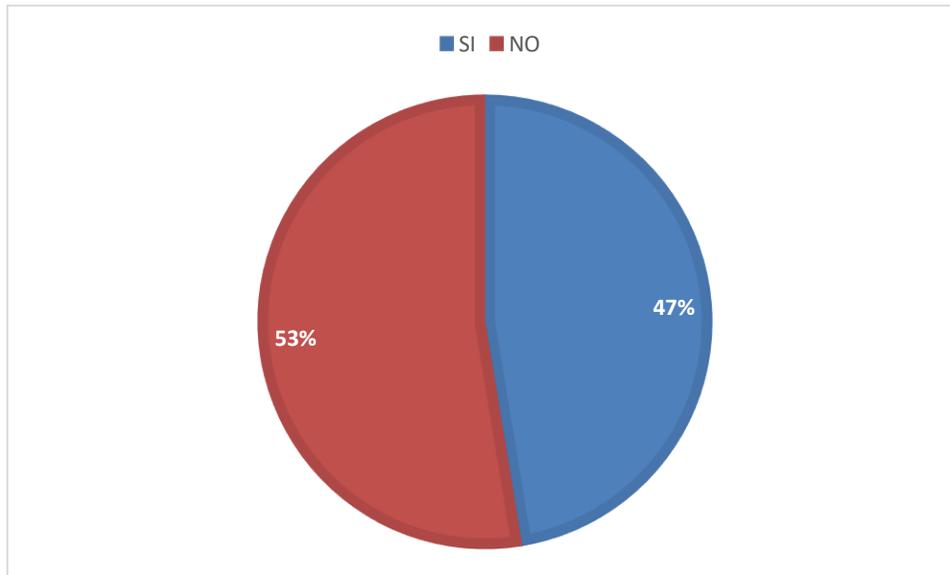


Figura 11: Porcentaje de médicos o personal de la salud

Como lo adelantamos al comienzo de este análisis, pudimos obtener las suficientes encuestas respondidas para dar con dos grandes grupos divididos a la mitad, tanto con el personal de la salud o médico como para personas comunes. Lo cual nos ayudaría a tomar un buen porcentaje de respuestas para ambos casos.

Para el personal de la salud, empezamos preguntando en qué tipo de establecimiento trabajan.

¿En qué tipo de establecimiento trabajas?

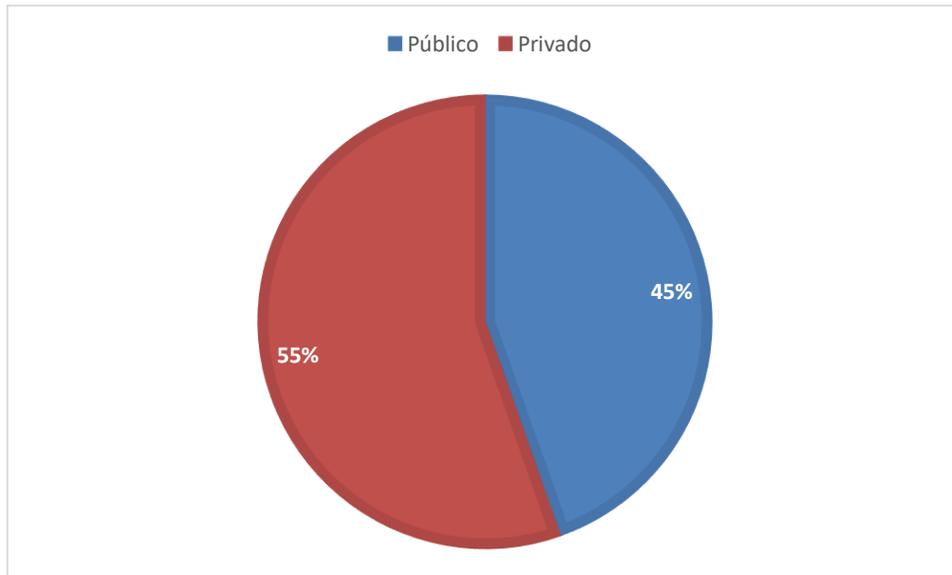


Figura 12: Porcentaje de lugar de trabajo, público o privado

Como habíamos comentado antes, con esta pregunta buscamos saber qué cantidad de personas trabajan en establecimientos privados, ya que estos, en su mayoría, tienen más probabilidad de contar con equipos de última tecnología a diferencia de los establecimientos públicos, los cuales dependen de la inversión del estado para mejorar su equipamiento.

¿Conoces el significado de las siglas IoT?

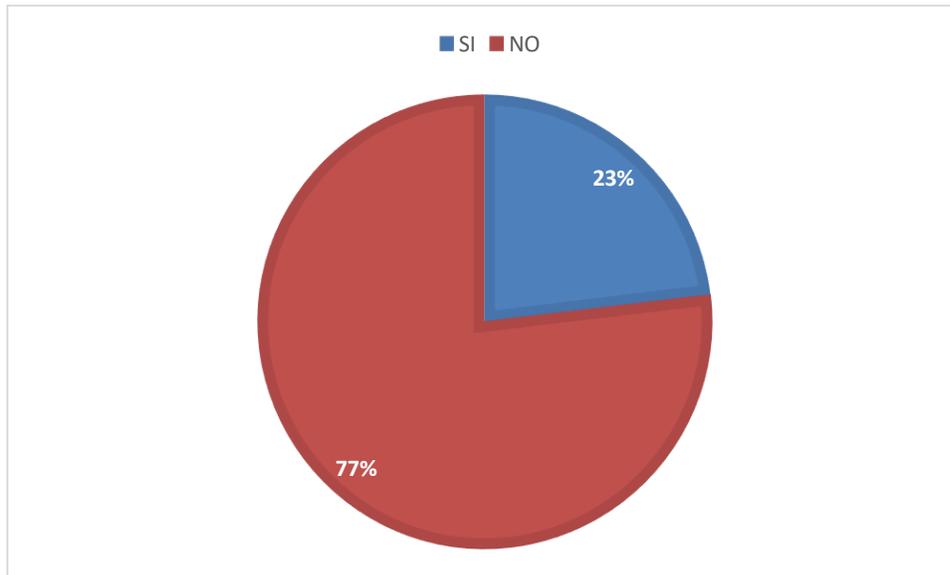


Figura 13: Conoces las siglas “IoT”

Aunque el “IoT” tiene varios años ya en el mercado, no es una tecnología muy conocida por el consumidor regular, aun así, pudimos observar que el personal médico o de la salud cuenta con un 23% de conocimiento del tema, a diferencia de las personas ajenas, para las cuales el porcentaje es mucho menor. A su vez, es probable que estos médicos cuenten con alguna herramienta o dispositivo que esté relacionado con esta tecnología, lo cual les hizo más fácil identificar las siglas.

¿En tu lugar de trabajo hay algún equipo IoT?

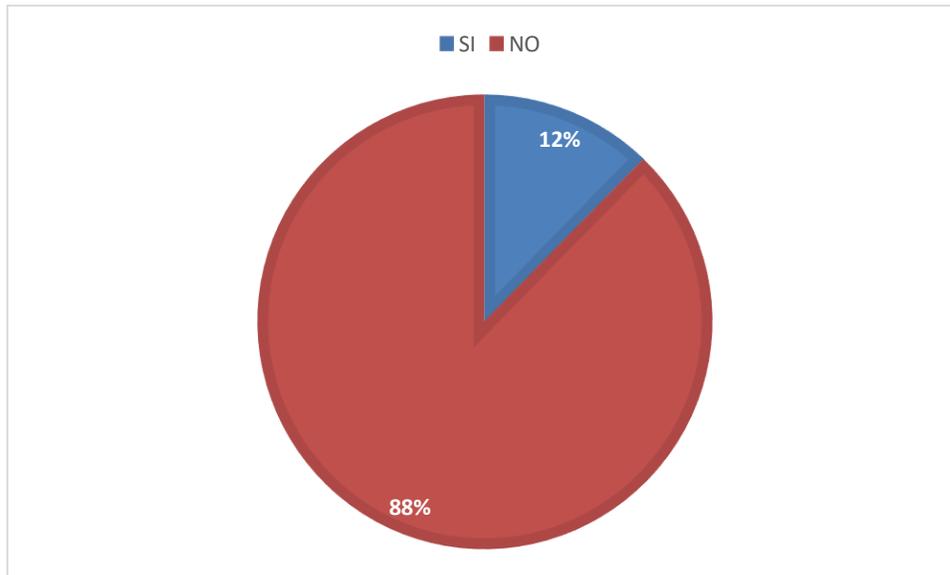


Figura 14: Trabajas con algún equipo IoT

Ya obteniendo poco porcentaje de conocimiento de la tecnología, más difícil aún fue obtener que algún médico trabajará con esta tecnología, sin embargo, obtuvimos varias respuestas positivas como es la telemedicina, muy utilizada hoy en día en consecuencia principalmente por la pandemia en la que nos encontramos.

A su vez, obtuvimos respuestas negativas, y en su mayoría, por parte de los médicos que prestan servicios en hospitales o establecimientos públicos, los cuales comentaron que no cuentan con WI-FI en sus instalaciones, creemos que esto se debe a que la mayoría de nuestros encuestadores se encuentran en países subdesarrollados que no cuentan con los mejores servicios en todas la áreas posibles.

Una vez fueron consultadas las preguntas básicas, se dio la explicación de en qué se basa nuestro proyecto y que puede ofrecer, para que nos pudieran contestar dos últimas preguntas las cuales nos servirían para completar o mejorar nuestros parámetros.

“Nuestro sistema está basado en la monitorización eficiente de la salud del paciente en Cloud, cuando el paciente no se encuentra en el centro de salud. Este sistema les permitirá a los médicos controlar los parámetros fisiológicos normales para cada paciente de manera online y poder tomar las medidas necesarias en caso de emergencia. En este proyecto se implementa un sistema de monitoreo de parámetros fisiológicos basado en el microprocesador NodeMCU y un servidor Cloud (Ubidots). Varios parámetros básicos como ECG, la presión arterial, oxígeno en la sangre, la temperatura corporal entre otros, serán sensados y procesados para poder visualizarlos ya sea en la pantalla LCD del aparato como en el Cloud al que tiene ingreso el médico o personal de la salud a distancia.”

¿Los parámetros agregados son suficientes para monitorear la salud de un paciente a distancia?

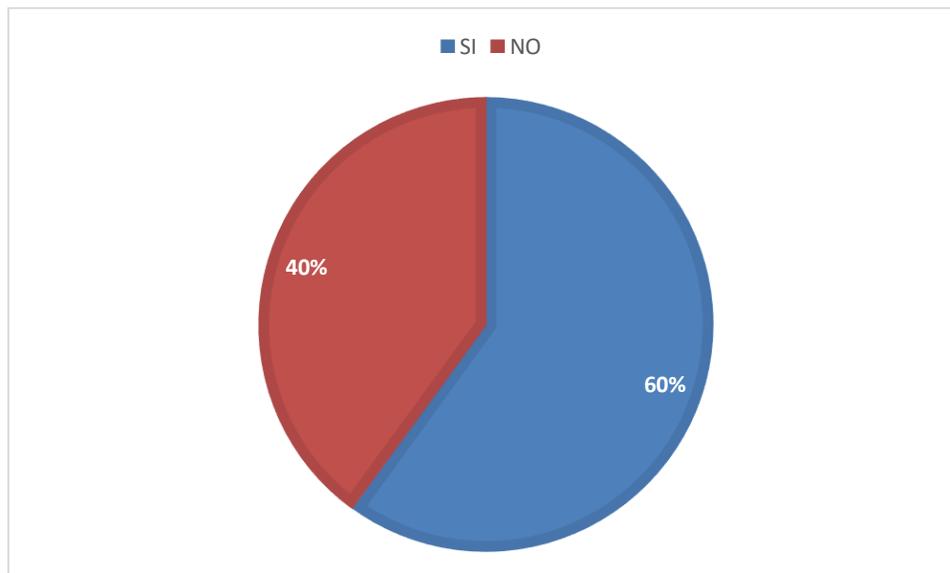


Figura 15: Parámetros medidos al paciente

Es evidente, que dependiendo de la situación del paciente o la urgencia, los parámetros básicos que estamos midiendo no son suficientes, por lo cual 40% de los médicos no estuvieron de acuerdo, ya que no abarca todo lo necesario para monitorear cualquier tipo de

urgencia o paciente sino algunos en especial dependiendo de su estado, además que hoy en día la tecnología IoMT no cuenta con una extensa gama de sensores, sino que por lo contrario son muy puntuales los que han sido desarrollados, más aún, no todos los que existen cuentan con el porcentaje de error o confianza que necesita un médico para hacerle un correcto seguimiento al paciente a distancia.

Con lo que podemos concluir que hoy en día, la brecha entre la medicina clásica y las nuevas tecnologías es muy grande, sin embargo es probable que en los próximos años esto vaya cambiando y creciendo de manera exponencial.

¿Qué otro parámetro le gustaría medir?

Para esta pregunta obtuvimos respuestas de todo tipo, muchos quedaron satisfechos con los parámetros medidos para poder hacer un seguimiento básico, algunas otras sugerencias salen del rango de evaluaciones que se le podría hacer a un paciente a distancia como lo es “RX Tórax”, pero nos quedamos con una muy particular, que la hora de tener un paciente hospitalizado es importante seguir y es la Escala de Glasgow.

“La Escala de Coma de Glasgow (en Inglés Glasgow Coma Scale (GCS)) es una escala de aplicación neurológica que permite medir el nivel de conciencia de una persona. Una exploración neurológica de un paciente con traumatismo craneoencefálico debe ser simple, objetiva y rápida.

*La Escala de Coma de Glasgow utiliza tres parámetros que han demostrado ser muy replicables en su apreciación entre los distintos observadores: **la respuesta verbal, la respuesta ocular y la respuesta motora.**”* Generación Elseiver. (5 de junio de 2017). <https://www.elsevier.com/es-es/connect/medicina/escala-de-coma-de-glasgow>

Es una evaluación que necesita hacer el médico al tener un paciente en un estado comprometido, sin embargo, creemos podría solucionarse con una cámara y con ayuda de una persona que acompañe al paciente y así evaluar las respuestas a estas pruebas. Queda además

confirmar, que nuestro equipo, está diseñado para una evaluación y seguimiento básico, queda descartado para pacientes en estado crítico o en coma.

Para culminar, quisimos evaluar la aceptación del equipo o tecnología por parte del paciente o familiar de este, consultando varias preguntas básicas.

¿Conoces las siglas IoMT?

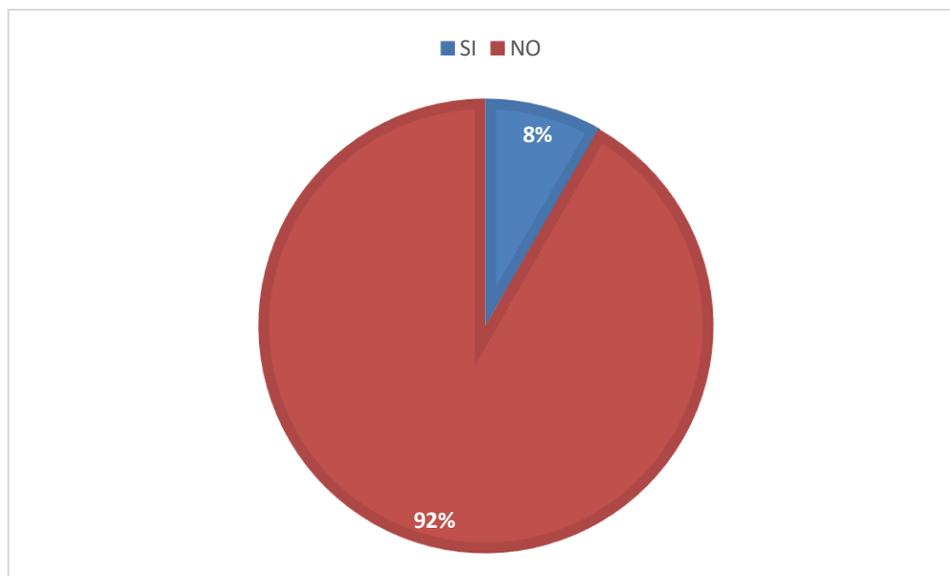


Figura 16: Conoces las siglas IoMT

Como habíamos mencionado antes, para el personal médico era un poco más conocido este término, sin embargo, para el ciudadano común o paciente, son siglas casi irreconocibles. Hoy en día, es muy probable que el área de IoT más conocida sea la domótica, de la mano de productos como lo son Alexa de Amazon, Google Home, entre otros. Sin embargo, estos equipos son más conocidos en países como Estados Unidos donde la tecnología llega a toda la población a pesar de sus costos.

Al explicar en que se basa nuestro proyecto, al igual que lo hicimos con el personal médico, quisimos evaluar el nivel de confianza en el equipo o tecnología para la sociedad.

¿Te gustaría hacer uso de esta tecnología?

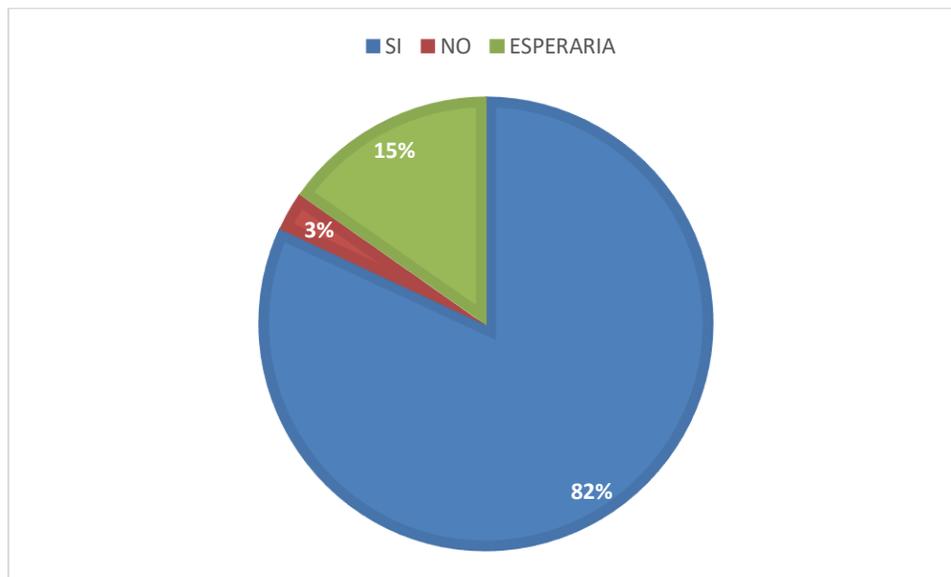


Figura 17: Te gustaría usar esta tecnología

Es evidente que la comodidad del paciente es uno de los aspectos más importantes, pasar horas de más en las habitaciones, o tener que dormir en lugares incómodos no es del agrado de nadie. Por lo cual esta nueva oportunidad es vista con buenos ojos, sin embargo, existe un 18% de personas que no les gustaría o que directamente esperarían a que la tecnología esté mucho más desarrollada y les pueda dar la confianza suficiente.

¿Si un familiar está en estado crítico, ¿dejarías que sea evaluado mediante esta forma?

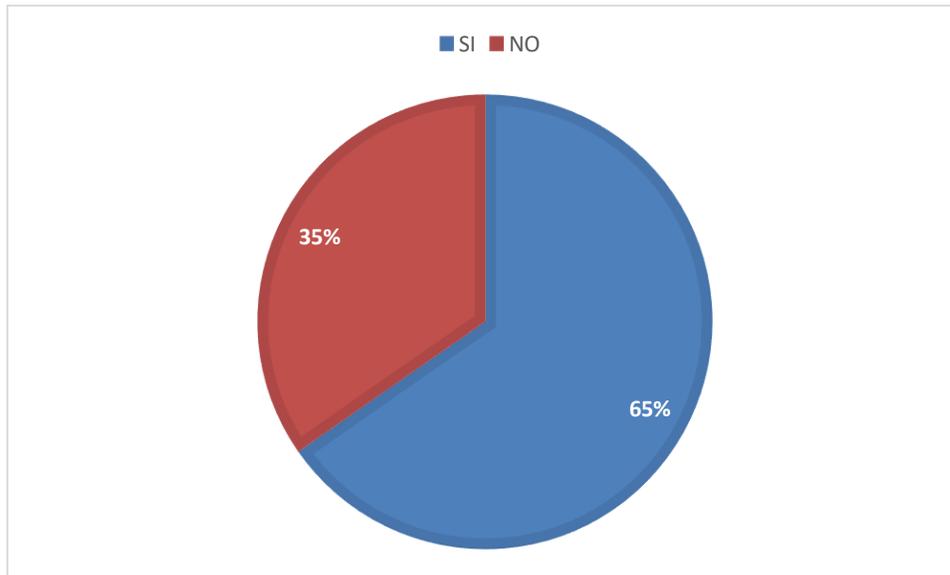


Figura 18: Dejarías que un familiar en estado crítico sea observado de esta manera

Aunque como hemos dicho anteriormente, nuestro proyecto no está diseñado para el monitoreo de personas en estado crítico, quisimos saber qué personas seguirán confiando en esta nueva tecnología cuando se trate de un pariente cercano quien la esté usando. Lo que evidentemente nos llevó a bajar un poco la confianza en esta tecnología y oportunidad, ya que es algo que no está perfectamente puesto en funcionamiento y las personas prefieren no arriesgar.

3. METODOLOGÍA DE DESARROLLO

Considerando la problemática comentada anteriormente, y el análisis de la encuesta en cuanto a los parámetros fisiológicos necesarios para poder monitorear un paciente a la distancia, se seleccionaron los sensores y hardware pertinente para su mejor uso, así como también la plataforma a utilizar para observar los resultados de los mismos.

El prototipo a desarrollar contará con un total de cinco sensores para medir algunos de los parámetros más solicitados así como algunas otras funciones como la estabilidad del paciente y su ubicación, para subir la información obtenida por los sensores utilizaremos el microprocesador NodeMCU, ya es que este nos permite conectarnos a los sensores mediante comunicación I2C y a su vez cuenta con módulo Wi-Fi que enviará la información directamente a la plataforma seleccionada, esta sería Ubidots una plataforma IoT que cuenta con un servicio pago y uno gratuito para recolectar datos en la nube y tiene interfaces seleccionable para mostrar los datos de la manera que más nos convenga.

Por lo que básicamente tendremos un conjunto de sensores conectados a un microprocesador que se encuentran en la casa del paciente que lo necesite, este se conectara a internet mediante el “cerebro” del equipo que será el NodeMCU, y enviará la información sensada a la nube que podrá ser vista por el médico o personal de la salud pertinente a través de la plataforma Ubidots, que cuenta con un interfaz gráfico diseñado principalmente para la tecnología IoT.

3.1. Microprocesador ESP8266 NodeMCU

NodeMCU es una plataforma de código abierto ampliamente difundida en las aplicaciones de IoT. El firmware está basado en el lenguaje Lua y el hardware se basa en el módulo ESP-12. El firmware que incluye es el mismo que es ejecutado en el ESP8266, por lo que la funcionalidad requerida anteriormente es cumplida de la misma manera, pero en un kit de desarrollo que concentra la interfaz de programación y el módulo Wi-Fi en un solo componente.



Figura 19: ESP8266 NodeMCU

Tiene potentes capacidades de procesamiento y almacenamiento que le permiten integrarse con sensores y dispositivos específicos de aplicación a través de sus GPIOs con un desarrollo mínimo y carga mínima durante el tiempo de ejecución. Su alto grado de integración en el chip permite una circuitería externa mínima, y la totalidad de la solución, incluyendo el módulo está diseñado para ocupar el área mínima en un PCB. Además de alimentarse por USB, se puede energizar aplicando un voltaje de 5V de CC a uno de sus pines. Del mismo modo, permite alimentar dispositivos externos con 3.3V a través de uno de sus pines de salida. Se puede observar el detalle de sus pines en la siguiente figura:

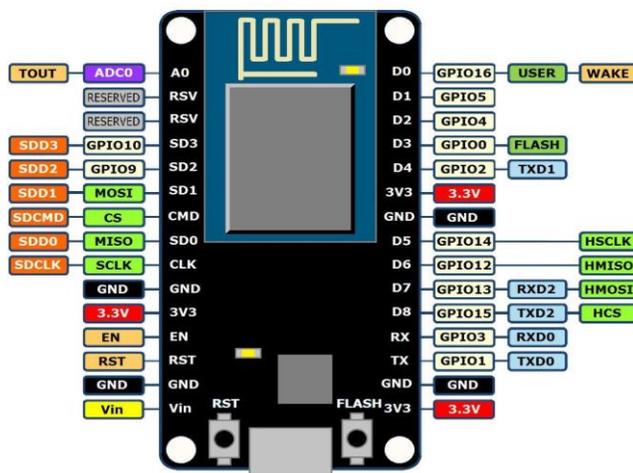


Figura 20: Diagrama de pinout de NodeMCU

3.2. Sensores

Un sensor es un dispositivo o módulo que tiene como propósito detectar eventos o cambios en un ecosistema y enviar la información a dispositivos electrónicos, frecuentemente a una computadora. Un sensor es siempre usado con otros dispositivos electrónicos, algunos tan simples como un Led o tan complejos como una computadora.

Los sensores se utilizan en objetos cotidianos como botones de ascensor y lámparas que se atenúan o se iluminan al tocar la base, además de innumerables aplicaciones de las que la mayoría de las personas imaginan. Con los avances en micro maquinaria y plataformas de micro controladores fáciles de usar, los usos de los sensores se han expandido más allá de los campos tradicionales de medición de temperatura, presión o flujo. Además, los sensores analógicos como los potenciómetros y las resistencias de detección de fuerza todavía se utilizan ampliamente. Las aplicaciones incluyen fabricación y maquinaria, aviones y aeroespacial, automóviles, medicina, robótica y muchos otros aspectos de nuestras corporaciones de la vida cotidiana.

3.3. Sensor de electrocardiograma AD8232



Figura 21: ECG Sensor AD8232

El Electrocardiograma (ECG) es el proceso de registrar la actividad eléctrica del corazón durante un período de tiempo utilizando electrodos colocados sobre la piel. Estos electrodos detectan los pequeños cambios eléctricos en la piel que surgen del patrón electrofisiológico del músculo cardíaco de despolarización y repolarización durante cada latido del

corazón. Es una prueba de cardiología que se realiza con mucha frecuencia. En un ECG convencional de 12 derivaciones, se colocan diez electrodos en las extremidades del paciente y en la superficie del pecho.

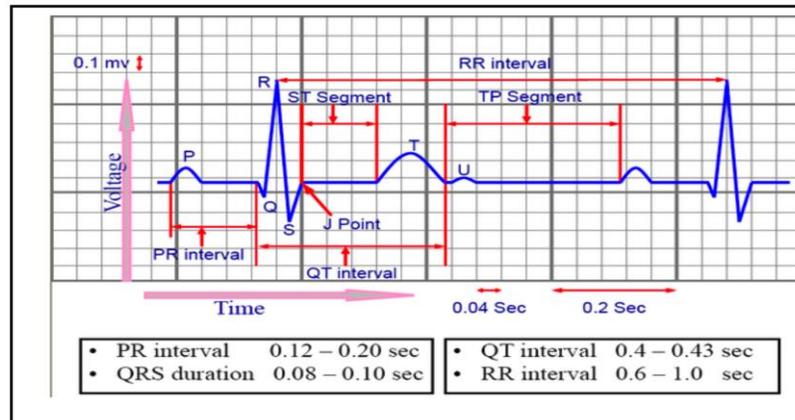


Figura 22: ECG Gráfica y explicación

La magnitud general del potencial eléctrico del corazón se mide luego desde doce ángulos diferentes ("derivaciones") y se registra durante un período de tiempo (generalmente diez segundos). De esta manera, la magnitud y la dirección generales de la despolarización eléctrica del corazón se capturan en cada momento a lo largo del ciclo cardíaco. La gráfica de voltaje versus el tiempo producido por este procedimiento médico no invasivo es un electrocardiograma.

Durante cada latido, un corazón sano tiene una progresión ordenada de despolarización que comienza con las células marcapasos en el nódulo sinoauricular, se extiende a través de la aurícula, pasa a través del nódulo auriculoventricular hacia el haz de His y las fibras de Purkinje, extendiéndose hacia abajo y hacia la izquierda a lo largo de los ventrículos. Este patrón ordenado de despolarización da lugar al trazado característico del ECG. Para el médico capacitado, un ECG transmite una gran cantidad de información sobre la estructura del corazón y la función de su sistema de conducción eléctrica. El monitor de frecuencia cardíaca de un solo cable SparkFun AD8232 es una placa que se utiliza para medir la actividad eléctrica del corazón. Esta actividad eléctrica se puede registrar como un ECG o electrocardiograma y emitirse como una lectura analógica. Los ECG pueden ser extremadamente ruidosos, el monitor de frecuencia cardíaca de un

solo cable AD8232 actúa como un amplificador operacional para ayudar a obtener una señal clara de los intervalos PR y QT fácilmente.

El AD8232 es un sistema integrado para realizar ECG y otras aplicaciones de medición de biopotencial. Está diseñado para extraer, amplificar y filtrar pequeñas señales de biopotencial en presencia de condiciones ruidosas, como las creadas por el movimiento o la colocación de electrodos. El monitor de frecuencia cardíaca AD8232 tiene conexiones a las que se le pueden soldar pines, cables u otros conectores. SDN, LO +, LO-, OUTPUT, 3.3V, GND proporcionan pines esenciales para utilizarlo con un Arduino u otra placa de desarrollo. También se proporcionan en esta placa los pines RA (brazo derecho), LA (brazo izquierdo) y RL (pierna derecha) para conectar y usar sus propios sensores personalizados. Además, hay una luz indicadora LED que pulsará al ritmo de los latidos del corazón. Se requieren almohadillas de sensor biomédicas y cable del sensor para usar el monitor cardíaco.

3.4. Sensor de frecuencia cardíaca y pulsioximetría MAX30102

El MAX30102 es un módulo integrado de pulsioximetría y monitor de frecuencia cardíaca. Incluye LED internos, foto detectores, elementos ópticos y electrónica de bajo ruido con rechazo de luz ambiental. Este integrado proporciona una solución de sistema completa para facilitar el proceso de diseño para dispositivos móviles y portátiles.

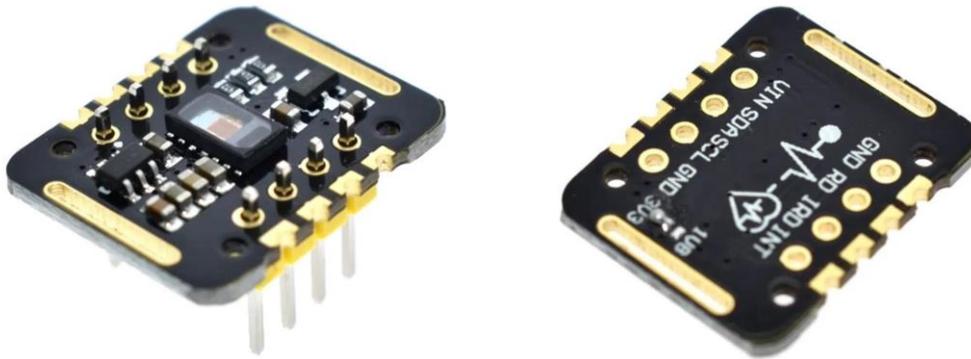


Figura 23: Sensor MAX30102

El sensor funciona con una única fuente de alimentación de 1,8 V y una fuente de alimentación independiente de 3,3 V para los LED internos. La comunicación se realiza a través de una interfaz estándar compatible con I2C. El módulo se puede apagar a través del software sin corriente de espera, lo que permite que los rieles de alimentación permanezcan alimentados en todo momento.

Beneficios y especificaciones:

- Monitor de frecuencia cardíaca y sensor de oxímetro de pulso en solución reflectante LED (Pequeño módulo óptico de 14 pines de 5,6 mm x 3,3 mm x 1,55 mm)
- Vidrio de cubierta integrado para una óptima y robusta actuación
- Operación de energía ultra baja para dispositivos móviles
- Frecuencia de muestreo programable y corriente LED para ahorro de energía
- Monitor de frecuencia cardíaca de bajo consumo (<1 mW)
- Corriente de apagado ultra baja (0,7 μ A, típico)
- Rango de temperatura de funcionamiento de -40 °C a + 85 °

3.5. Sensor de caída MPU6050



Figura 24: Sensor MPU6050

El módulo de sensor MPU6050 es un módulo completo de 6 ejes (acelerómetro de 3 ejes y giroscopio de 3 ejes). Este módulo utiliza sistemas microelectromecánicos (MEMS) que sirven para medir la aceleración, la velocidad, la orientación, el desplazamiento y muchos otros parámetros relacionados con el movimiento. Aparte de esto, también tiene un sensor de temperatura incorporado.

El módulo MPU6050 es de tamaño pequeño y consume poca energía. Aparte de esto, tiene una alta repetición, una alta tolerancia a los golpes y un bajo precio. Básicamente, el MPU6050 viene con una interfaz I2C y auxiliar I2C.

Características:

- Salida digital de 6 ejes.
- Giroscopio con sensibilidad de ± 250 , ± 500 , ± 1000 , y ± 2000 dps
- Acelerómetro con sensibilidad de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$
- Algoritmos embebidos para calibración
- Sensor de temperatura digital
- Entrada digital de video FSYNC
- Interrupciones programables

- Voltaje de alimentación: 2.37 a 3.46V
- Voltaje lógico: 1.8V±5% o VDD
- 10000g tolerancia de aceleración máxima

3.6. Sensor de ubicación GPS Module (GY-NEO6MV2)

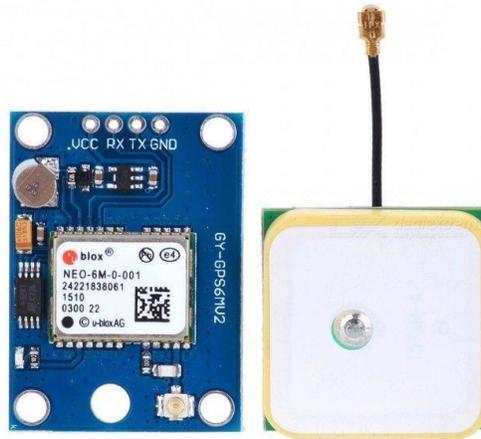


Figura 25: Sensor GY-NEO6MV2

El módulo GPS NEO6MV2 permite recibir señales GPS. Esto permite que un proyecto determine exactamente dónde se encuentra en la tierra, qué tan rápido se mueve, en qué dirección se mueve y cuántos satélites son visibles. La comunicación se realiza a través de una conexión serie TX / RX (solo se requieren 2 E/S).

Especificaciones:

- Precisión: 2,5 metros
- Antenna: Ceramic 25mm x 25mm (included)
- EEPROM puede guardar todos los datos de configuración
- LED indicador de señal

- Interface: TTL serial
- Velocidad por defecto: 9600 bps

3.7. Sensor de temperatura infrarrojo MLX90614

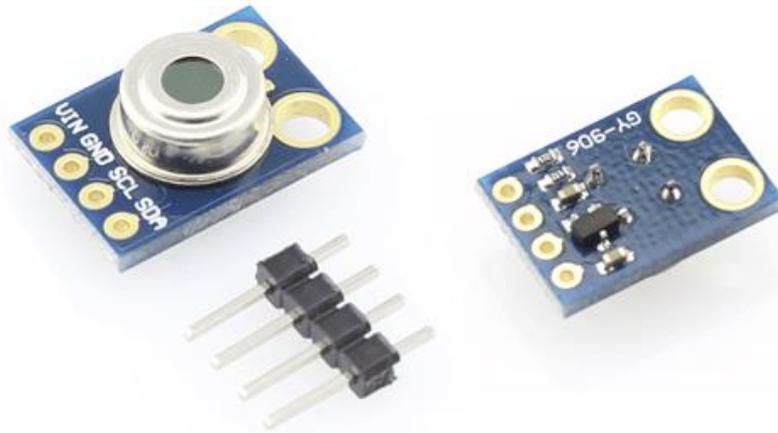


Figura 26: Sensor MLX90614

El Sensor de Temperatura infrarrojo MLX90614 es un chip de silicio con una fina membrana micro mecanizada, diseñada para ser sensible a la radiación infrarroja emitida por un objeto a distancia. El sensor posee una etapa de amplificación y digitalización de la señal procedente de la membrana. La salida del sensor es lineal y se compensa de acuerdo a las variaciones de la temperatura ambiente.

El sensor MLX90614 integra un circuito de filtrado de ruido, un conversor A/D de 17 bits de resolución y un procesador digital de señales, entregando un amplio rango de trabajo para objetos desde -70°C hasta 380°C , con una precisión de 0.5°C .

La salida del sensor es de tipo SMBus, que es muy similar al protocolo I2C, además se puede configurar una salida PWM de 10 bits.

ESPECIFICACIONES TÉCNICAS:

- Voltaje de Operación: 5V
- Rango de temperatura ambiente de trabajo: -40°C hasta +170°C
- Rango de temperatura de objeto: -70°C hasta +380°C
- Precisión: $\pm 0.5^\circ\text{C}$.
- ADC incorporado de 17 bits
- Protocolo de comunicación SMBUS (I2C)
- No necesita componentes adicionales

3.8. Servidor

En informática, un servidor es un programa informático o un dispositivo que proporciona funcionalidad para otros programas o dispositivos, llamados "clientes". Esta arquitectura se denomina modelo cliente-servidor. Los servidores pueden proporcionar varias funcionalidades, a menudo denominadas "servicios", como compartir datos o recursos entre varios clientes o realizar cálculos para un cliente. Un solo servidor puede servir a varios clientes y un solo cliente puede utilizar varios servidores. Un proceso de cliente puede ejecutarse en el mismo dispositivo o puede conectarse a través de una red a un servidor en un dispositivo diferente. Los servidores pueden ser servidores de bases de datos, servidores de archivos, servidores de correo, servidores de impresión, servidores web, servidores de juegos y servidores típicos de aplicaciones.

En la actualidad, los sistemas cliente-servidor se implementan con mayor frecuencia y a menudo se identifican con el modelo de solicitud-respuesta; un cliente envía una solicitud al servidor, que realiza alguna acción y envía una respuesta al cliente, generalmente con un resultado o con confirmación de recibo. La designación de una computadora como "hardware de clase de servidor" implica que está especializada para ejecutar servidores en ella. Esto a menudo implica que es más potente y confiable que las computadoras personales estándar, pero además, los grandes clústers de computación pueden estar compuestos por muchos componentes de servidores de fácil reemplazo.

El servidor es parte del modelo cliente-servidor; en este modelo, un servidor sirve datos para los clientes. La naturaleza de la comunicación entre un cliente y un servidor es Request and Response (solicitud y respuesta). Esto contrasta con el modelo “Peer to Peer” (P2P) en el que la relación es bajo demanda. En principio, cualquier proceso computarizado que pueda ser usado o llamado por otro proceso (particularmente de forma remota, para compartir un recurso) es un servidor, y el proceso o procesos que llaman es un cliente. Por lo tanto, cualquier computadora de propósito general conectada a una red puede albergar servidores. Por ejemplo, si algún proceso comparte archivos en un dispositivo, ese proceso es un servidor de archivos. De manera similar, el software del servidor web puede ejecutarse en cualquier computadora, por lo que una computadora portátil o una computadora personal pueden albergar un servidor web. Si bien Request-Response (solicitud-respuesta) es el diseño cliente-servidor más común, existen otros, como Publish-Subscribe (publicación-suscripción). Cuando se hace referencia al hardware, la palabra servidor generalmente designa modelos de computadora especializados para su función. En general, un servidor desempeña su función mejor que una computadora personal genérica.

3.9. Almacenamiento de Datos

Los datos que se recopilaban de los sensores a través del NodeMCU deben almacenarse en algún lugar para analizarlos. Para ello se utiliza en esta tesis Ubidots, un sitio web que permite almacenar datos y analizarlos. El NodeMCU como se mencionó anteriormente tiene la capacidad de conectarse con Internet y almacenar los datos necesarios en el servidor Ubidots. El NodeMCU recibirá datos de los sensores y con la ayuda de su conexión Wi-Fi, enviará los datos adquiridos al servidor Ubidots.

Para diferentes datos, los diferentes tipos de canales se crean en Ubidots. La temperatura del paciente se puede cargar continuamente a Ubidots con el NodeMCU. Desde el sensor de temperatura, la temperatura del paciente se guardará en un canal y también se almacenará la temperatura más alta medida. De esta manera podemos alertar al médico en caso de alta temperatura en el paciente.

Lo mismo ocurre con los datos del sensor

3.10. UBIDOTS

Ubidots es un creador de aplicaciones de Internet de las cosas (IoT) con análisis y visualización de datos. Convierte los datos de los sensores en conocimientos importantes para las decisiones comerciales, las interacciones de máquina a máquina y la investigación educativa. Los Ubidots son una forma fácil y asequible de integrar el poder de IoT en su negocio o investigación. La tecnología y el conjunto de ingeniería de Ubidots proporcionan a los usuarios una experiencia segura y sin complicaciones. Las API compatibles con dispositivos (a las que se accede a través de HTTP / Protocolos MQTT / TCP / UDP) proporcionan una conexión fácil y segura para enviar y recuperar datos en tiempo real desde y hacia el servicio en la nube. La plataforma de soporte de aplicaciones Ubidots admite visualización de datos interactiva en tiempo real (widgets) y un generador de aplicaciones de IoT que permite a los desarrolladores ampliar la plataforma con su propio código HTML / JS cuando sea necesario para la personalización privada. Los Ubidots están disponibles para que empresas e investigadores vinculan fácilmente computadoras, administren datos y guarden un área.

3.11. IFTTT

IFTTT (If This Then That) es un servicio basado en web mediante el cual podemos crear cadenas de declaraciones condicionales, llamadas applets. Usando estos applets, podemos enviar correos electrónicos, publicaciones en Twitter, publicaciones en Facebook, reproducir música, SMS, notificaciones, etc. Aquí en este proyecto, estamos usando IFTTT para enviar notificaciones mediante GMAIL al teléfono móvil cuando el sistema detecta una caída.



Figura 27: Mapa mental de la página web IFTTT

En este apartado, después de implementar el código, discutiremos cómo se muestra en la cuenta de Ubidots de un usuario y cómo se envía la reacción o los comentarios al médico. Los datos que obtuvimos luego de la transmisión se guardan en la nube del servidor Ubidots y desde allí con la ayuda de las funciones provistas por Ubidots podemos mostrar datos y enviar mensajes de alarma.

4. PRUEBAS REALIZADAS

4.1. Funcionamiento del ECG AD8232

A continuación se muestra un diagrama del circuito para conectar el sensor de ECG AD8232 con la placa NodeMCU ESP8266. Hay 6 pines en la placa de conexiones AD8232. El SDN no está conectado.

Conexión del módulo

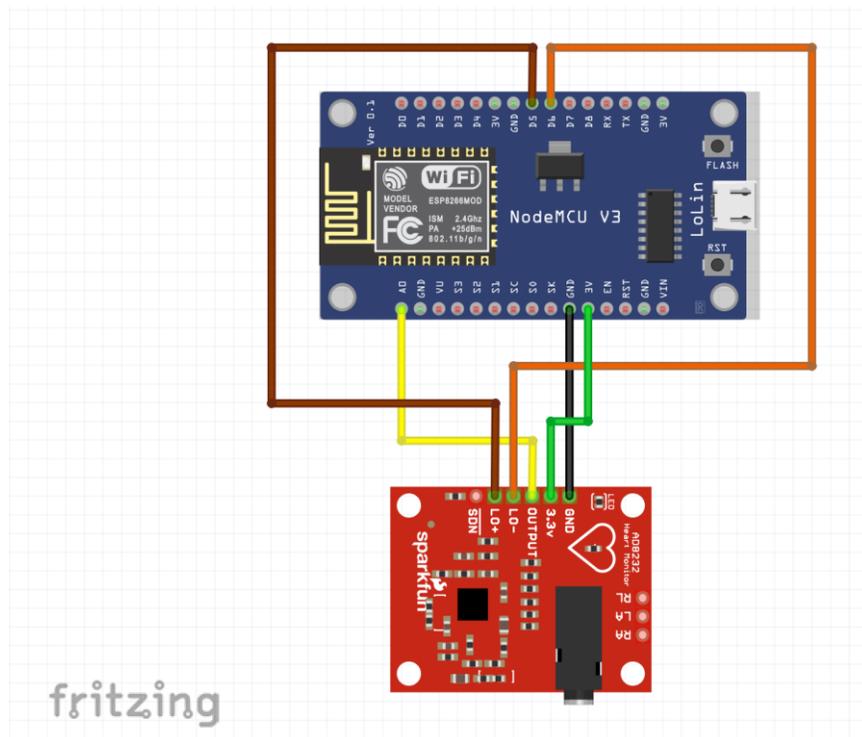


Figura 28: Esquema de conexión AD8232

Como se muestra en la figura anterior la SALIDA del AD8232 se conecta con la entrada analógica A0 de NodeMCU. Se Conecta LO + y LO- a D5 y D6 de NodeMCU respectivamente. Además se energiza el AD8232 con 3.3V VCC y se conecta su GND a GND.

Ubicación de los electrodos y calidad de la forma de onda

Para tener una mejor comprensión de los sistemas de monitorización de ECG, se realizó una investigación de antecedentes sobre la ubicación de los electrodos y la calidad de la forma de onda del ECG. El sistema de ECG tiene dos o más electrodos que se utilizan para controlar el voltaje en una o más derivaciones. El estándar clínico utiliza un ECG de 12 derivaciones, cada uno de los cuales proporciona una vista de la actividad eléctrica del corazón desde diferentes vistas. Cada derivación corresponde a un vector de potencial eléctrico de ECG. Los electrodos se colocan típicamente en una configuración conocida como triángulo de Einthoven, que está formado por los electrodos de detección del brazo derecho (RA), el brazo izquierdo (LA) y la pierna izquierda (LL), como se muestra en la Figura a continuación. Éstos son la base del eje frontal.

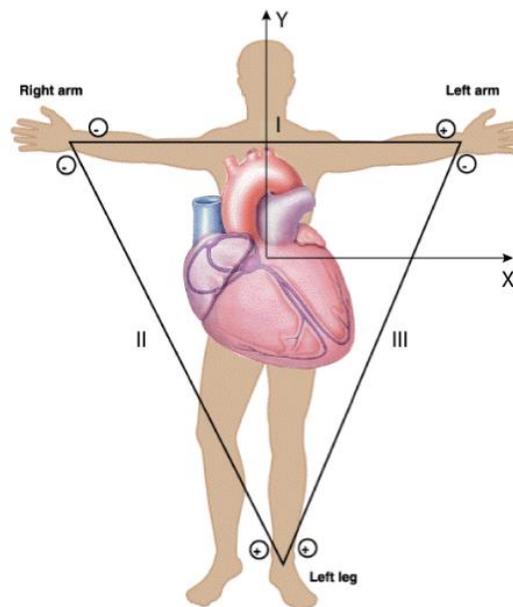


Figura 29: Triángulo de Einthoven

Una ubicación diferente de la derivación corresponde a una señal de ECG diferente, como se muestra en la Figura a continuación. La derivación II se usa comúnmente ya que proporciona la onda R positiva más grande que es importante para medir la frecuencia cardíaca.

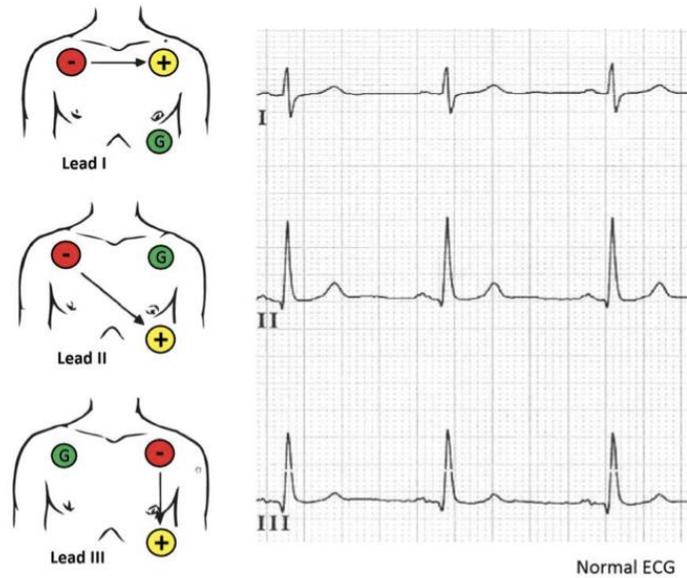


Figura 30: Ubicación de almohadillas y salida de ECG

Esta configuración permitiría seleccionar una derivación I, una derivación II o una derivación III. Los dos electrodos se utilizan para formar un vector, mientras que el tercer electrodo es la referencia. Para tres electrodos, es posible tener hasta tres vectores únicos. Esta implementación tiene la ventaja de tener menos ruido y más opciones de selección de conductores. La primera opción requeriría tres electrodos ubicados alrededor del corazón. Tiene vistas de Lead I, II y III. Una desventaja es la facilidad de uso, ya que podría tener que cruzar los músculos pectorales y el tejido mamario. Otra alternativa tiene como objetivo brindar un mayor confort al paciente. La compensación limitaría la vista del lead, ya que solo tiene una buena vista del Lead II.

En la imagen a continuación se puede observar la representación virtual de la actividad eléctrica del corazón en función del tiempo, donde se observan los claros picos del comportamiento y resultados al medir la actividad en el paciente.

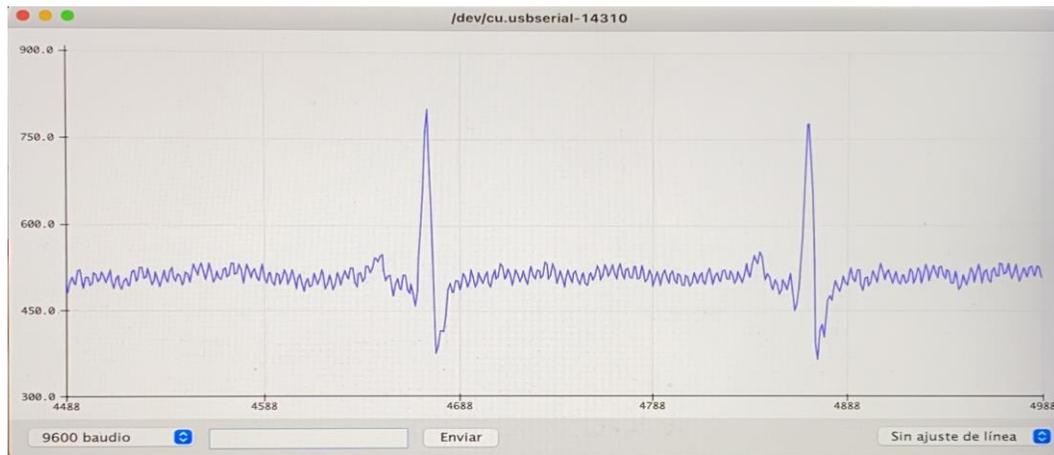


Figura 31: Gráfica prueba ECG

Según lo investigado, un electrocardiograma debería seguir la siguiente gráfica para un paciente en condiciones normales.



Figura 32: Electrocardiograma en perfecto funcionamiento

Comparando la imagen que nos da nuestro sensor en el monitor serie del IDE de Arduino junto con la de un electrocardiograma normal, nos damos cuenta que si bien no son iguales y no tiene todos los detalles necesarios, se asemejan mucho y más que todo en las frecuencias pico.

La diferencia se debe a que nuestro sensor es de baja calidad y precisión, tanto como los electrodos utilizados que son de un solo uso pero los utilizamos más de una vez, los que nos da

como resultado un electrocardiograma bastante parecido a uno real pero con pequeños detalles aceptables para nuestro prototipo.

La siguiente figura, muestra como el NodeMCU confirma su conexión a internet y al Ubidots, para luego empezar a enviar y publicar los datos necesarios para la representación gráfica de la actividad eléctrica del corazón del paciente.

```

.....
17:36:40.672 -> WiFi Connected
17:36:40.672 -> IP address:
3 17:36:40.672 -> 192.168.0.236
17:36:40.672 -> Attempting MQTT connection...
b 17:36:41.073 -> Connected
n 17:36:41.073 -> Publishing data to Ubidots Cloud
17:36:41.119 -> Publishing data to Ubidots Cloud
17:36:41.119 -> Publishing data to Ubidots Cloud
* 17:36:41.119 -> Publishing data to Ubidots Cloud
  
```

Figura 33: Salida del monitor serie para el envío de datos del ECG

Como habíamos comentado anteriormente, existen varias formas de conectar los electrodos al cuerpo humano para obtener los datos del corazón, sin embargo intentado las 3 mencionadas ninguna fue más eficiente que la que se puede observar en la siguiente imagen.

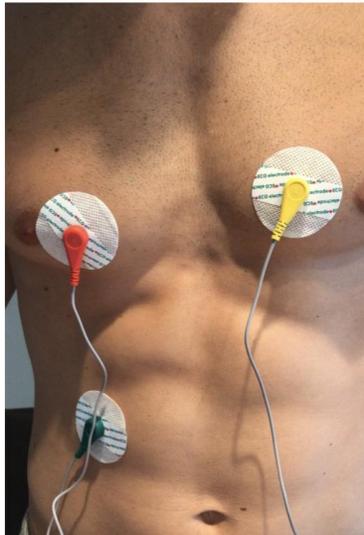


Figura 34: Ubicación elegida para almohadillas ECG

Con esta ubicación elegida para los electrodos, obtuvimos la mejor toma de datos y visualización de los mismos tanto en el monitor serie del IDE de Arduino como en el dashboard del Ubidots.

Por último, mostramos a continuación la forma en que la actividad cardiaca del paciente monitoreado es visualizada a través de la plataforma Ubidots por el médico o personal de la salud correspondiente.

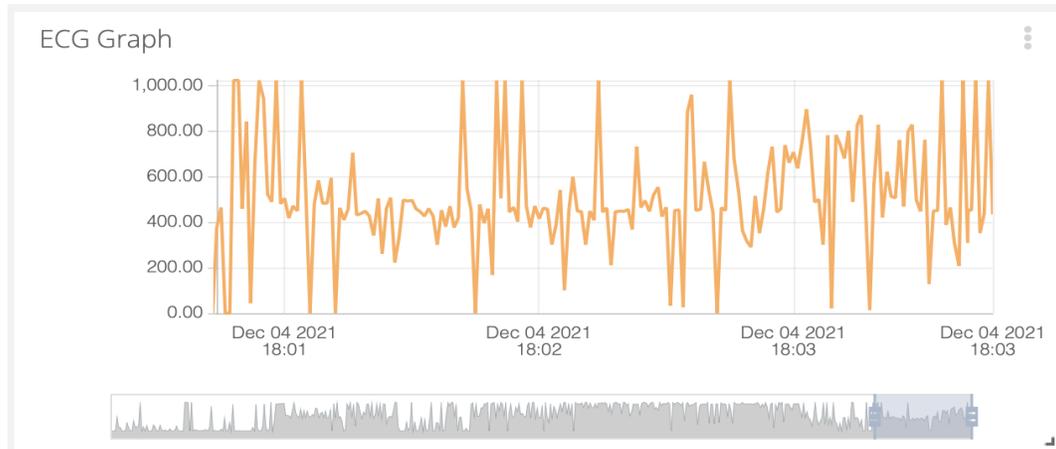


Figura 35: Electrocardiograma del paciente en la pantalla de Ubidots

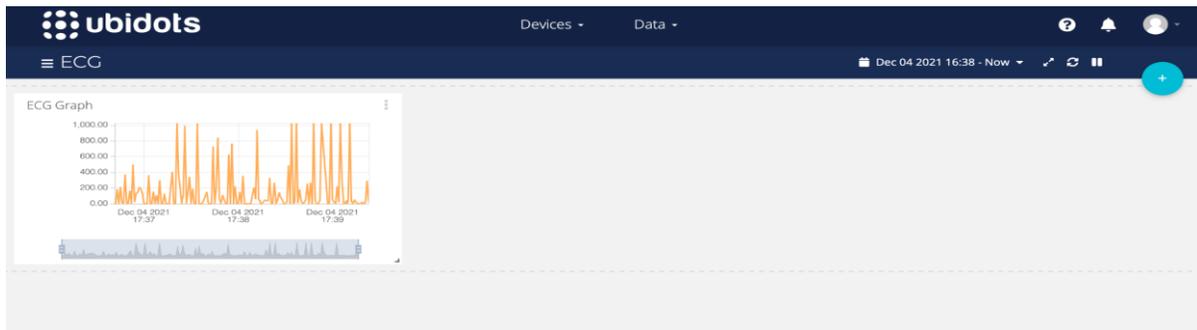


Figura 36: Dashboard del electrocardiograma del paciente

Se puede ver claramente una diferencia con la salida del monitor serie del IDE de Arduino, esto es debido a los valores que toma como referencia en el eje X y eje Y el dashboard de Ubidots, lo que nos acota la visualización más exacta del gráfico a detalle y se puede ver claramente que los pulsos están más unidos pero con una referencia de picos y valores altos parecida.

4.2. Funcionamiento del MLX90614

Conectamos el sensor de temperatura IR MLX90614 con NodeMCU ESP8266 y pantalla OLED.

Conexión del módulo

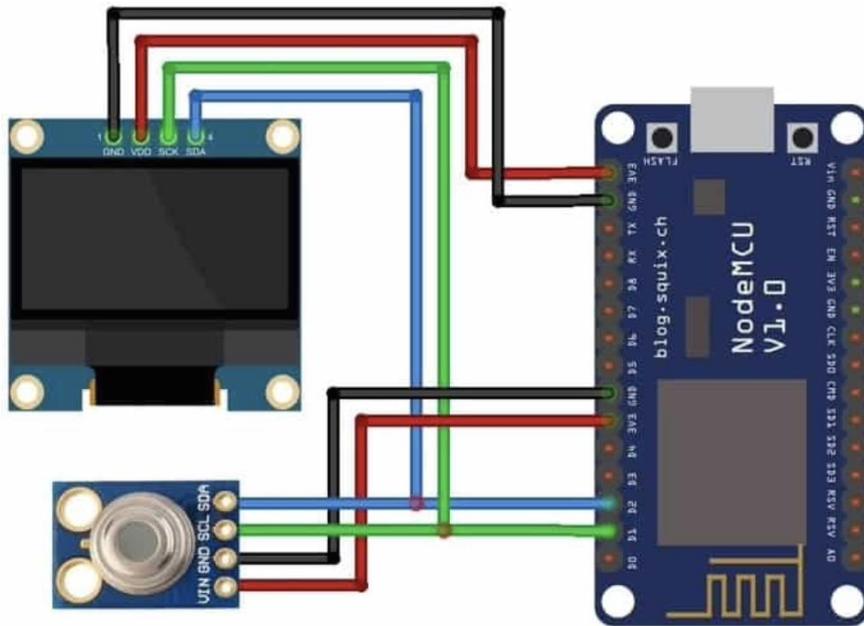


Figura 37: Esquema de conexión MLX90614-NodeMCU-OLED

Como se muestra en la figura anterior, conectamos el pin VCC de MLX90614 y la pantalla OLED al pin de 3.3V de NodeMCU ESP8266 y GND a GND. De manera similar, conectamos el pin SDA y SCL de MLX90614 a D2 y D1 de NodeMCU respectivamente.

Anteriormente fue mencionado que el sensor MLX90614 es utilizado para medir la temperatura corporal, hoy en día muy utilizados en las pistolas y equipos para entrar a cualquier establecimiento debido a la pandemia del COVID-19.

A continuación se muestra una imagen del monitor serie del IDE de Arduino, donde nos da el resultado de los datos obtenidos por el sensor, dando como resultado un promedio de 36 grados Celsius.

Como es sabido, la temperatura corporal varía entre los 35-37 grados Celsius en una persona promedio sin que presente alguna enfermedad o síntomas, por lo que obtuvimos un excelente respuesta del sensor aun sabiendo sus deficiencias como son la exactitud.

```

15:29:42.873 -> Ambiente = 27.41*C      Cuerpo = 35.69*C
15:29:42.927 ->
15:29:43.482 -> Ambiente = 27.47*C      Cuerpo = 35.75*C
15:29:43.528 ->
15:29:44.076 -> Ambiente = 27.51*C      Cuerpo = 36.05*C
15:29:44.129 ->
15:29:44.677 -> Ambiente = 27.59*C      Cuerpo = 36.05*C
15:29:44.731 ->
15:29:45.279 -> Ambiente = 27.65*C      Cuerpo = 36.05*C
15:29:45.333 ->
15:29:45.881 -> Ambiente = 27.75*C      Cuerpo = 36.07*C
15:29:45.935 ->
15:29:46.516 -> Ambiente = 27.81*C      Cuerpo = 36.07*C
    
```

Figura 38: Monitor Serie MLX90614

En la figura que tenemos a continuación, mostramos una captura de pantalla del Dashboard del Ubidots donde podemos ver a través de la interfaz gráfica la temperatura del cuerpo humano, si bien el resultado obtenido no era el esperado ya que no está en el rango de los valores aceptables, tiene un error menor al 5% lo que lo hace aceptable para nuestro prototipo.



Figura 39: Temperatura medida y observada mediante Ubidots

Por último, agregamos a la interfaz gráfica del Ubidots el otro valor medido que sería el ECG para mostrar cómo en una misma pantalla se pueden observar varios de los valores medidos por el equipo.

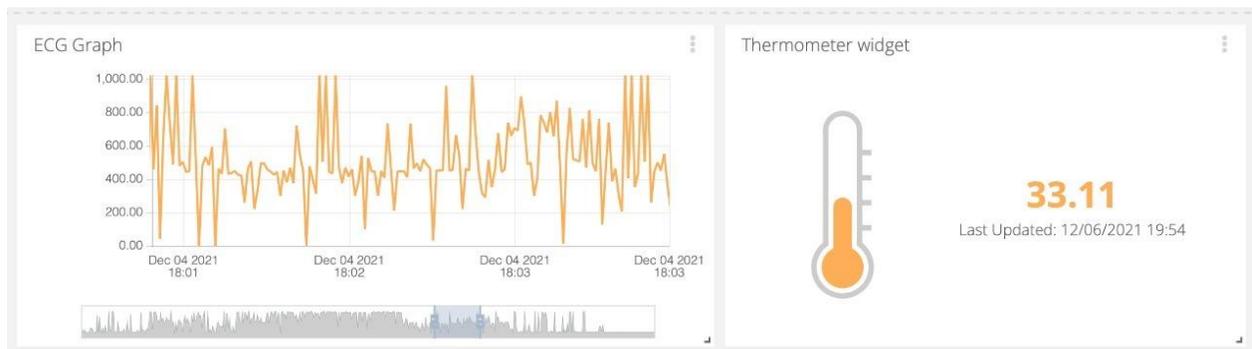


Figura 40: Dashboard del Ubidots con ECG y temperatura corporal

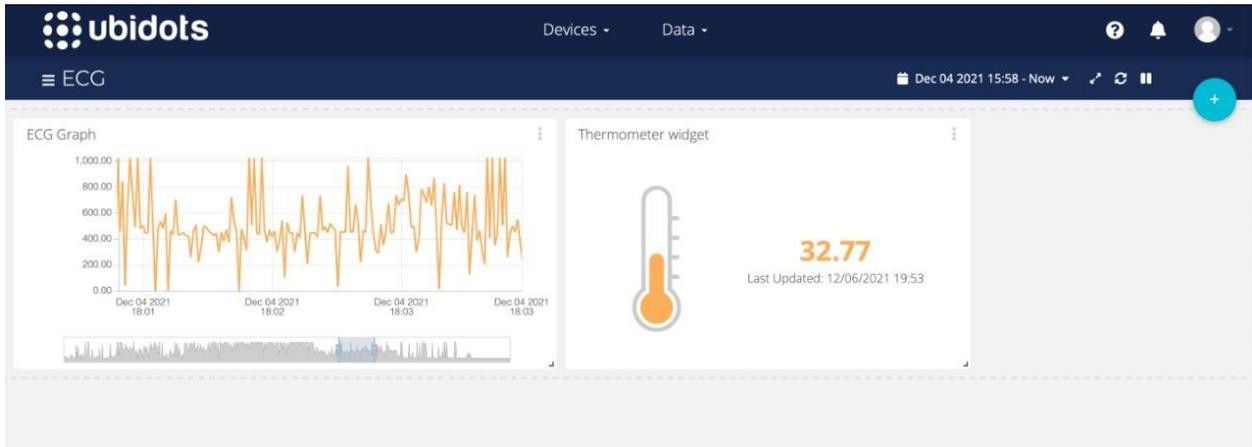


Figura 41: Dashboard completo del Ubidots con ECG y temperatura corporal

4.3. Funcionamiento del MAX30102

Conectamos el oxímetro de pulso MAX30102 con el NodeMCU ESP8266. El diagrama de circuito y la conexión se muestran a continuación.

Conexión del módulo

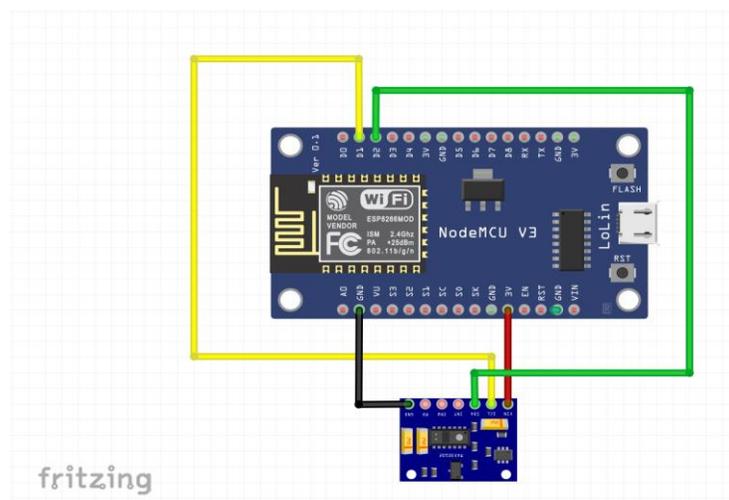


Figura 42: Esquema conexión MAX30102

Como se muestra en la figura anterior al MAX30102 se le conectan sus pines SDA a los pines D2 y SCL a D1 de la placa NodeMCU ESP8266. La fuente de alimentación requerida por el NodeMCU es 3.3V. Así que se conectó el terminal VCC a 3.3V de NodeMCU.

En la siguiente imagen, se puede observar el monitor serie del IDE de Arduino, donde se obtienen los valores medidos por el oxímetro de pulso y a su vez son enviados a través de internet al programa Ubidots.

```
-> Sending data...
-> Values sent by the device
-> , HR=100, SPO2=72, HR=100, SPO2=72, HR=100, SPO2=72, HR=100, SPO2=72,
-> payload:
-> {"Pulso":{"value":100},"% Oxígeno en la sangre":{"value":72}}
-> -----
->
-> Sending data...
-> Values sent by the device
-> , HR=100, SPO2=72, HR=100, SPO2=72, HR=100, SPO2=72, HR=100, SPO2=72,
-> payload:
-> {"Pulso":{"value":100},"% Oxígeno en la sangre":{"value":72}}
-> -----
```

Figura 43: Monitor serie con la salida de oxígeno en la sangre y pulso del paciente

Se puede observar de la imagen anterior los dos valores medidos, el porcentaje de oxígeno en la sangre y el pulso del paciente dándonos como resultados:

- Pulso: 100 latidos por minuto
- Oxígeno en la sangre: 72 mm Hg de oxígeno en la sangre

Una frecuencia cardiaca o pulso en reposo normal para adultos oscila entre 60 y 100, luego entran otros factores importantes a tomar en cuenta como lo es el peso, su condición física, las emociones y enfermedades, por lo cual el valor obtenido está entre lo esperado y aceptado para nuestro prototipo.

Para el oxígeno en la sangre en un paciente normal sin complicaciones varía entre 70 y 100 mm Hg en la sangre, por lo que entraría también en el rango de valores aceptables para nuestro prototipo.

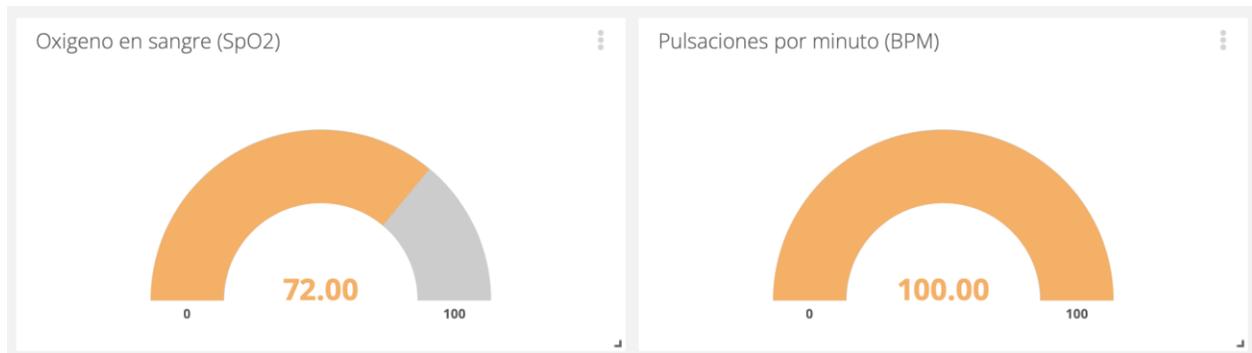


Figura 44: Valores de Oxígeno en la sangre y pulso del paciente

En la imagen anterior, podemos observar el dashboard del sensor estudiado en la plataforma Ubidots para ser observado por el médico o personal de la salud pertinente, los valores obtenidos están dentro de los rangos esperados para un paciente adulto en estado normal.

Por último, adjuntamos una imagen de como sería visto por el médico o personal de la salud capacitado el dashboard de la plataforma Ubidots, mostrando los valores capturados por los tres sensores a la misma vez, que serían ECG, oxímetro de pulso y temperatura corporal.

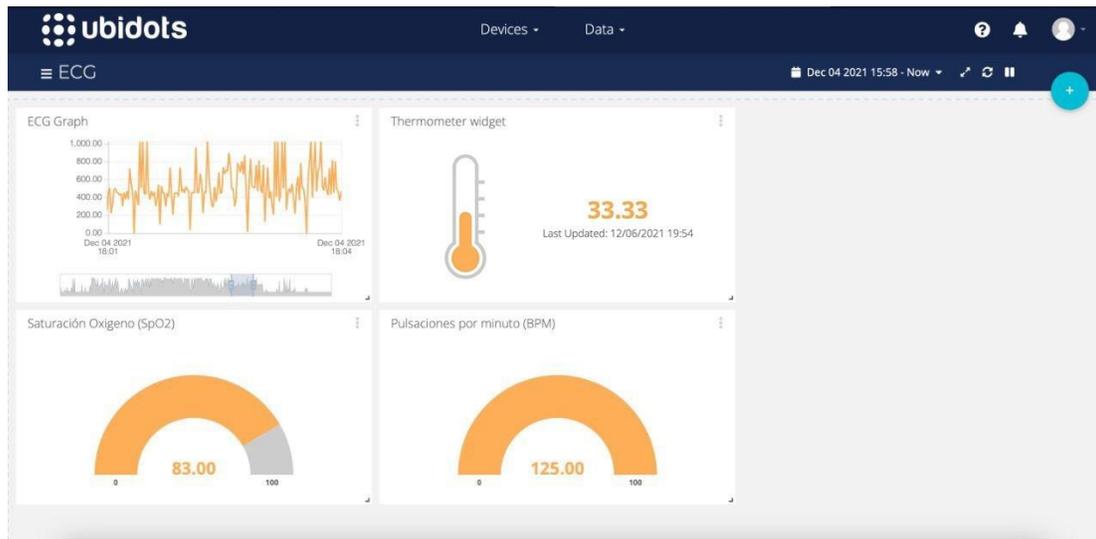


Figura 45: Dashboard completo de Ubidots

4.4. Funcionamiento del GY-GPS6MV2

Como habíamos comentado anteriormente, el sensor GY-GPS6MV2 se utiliza como módulo GPS, este nos ayuda a identificar a tiempo real la ubicación exacta del paciente y así al momentos de una emergencia poder enviar directamente la ambulancia o unidad de rescate a la localización, si el médico de turno al ir viendo los parámetros fisiológicos del paciente detecta una anomalía con urgencia, solo con buscar la ubicación del paciente podrá enviar o solicitar la ayuda pertinente para el caso según corresponda.

Conexión del módulo

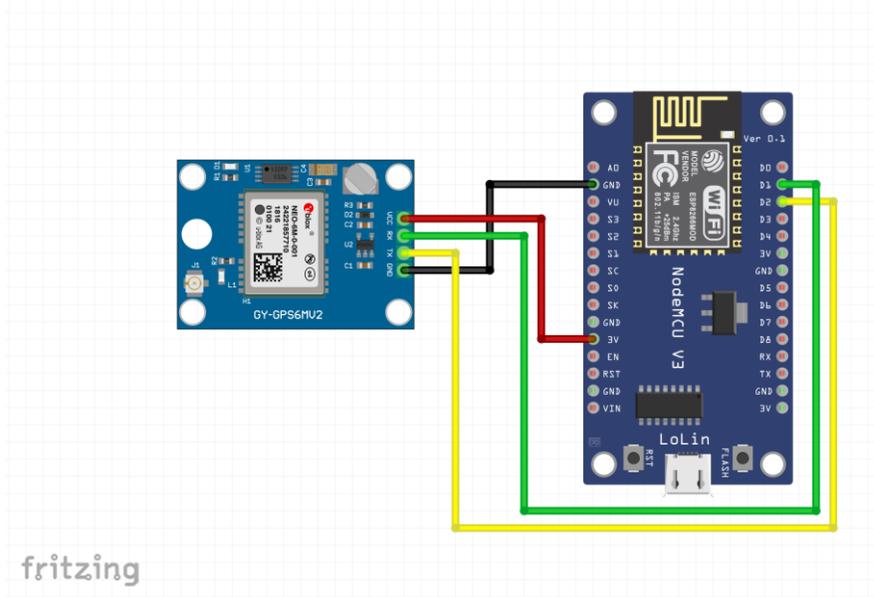


Figura 46: Esquema conexión GY-GPS6MV2 con NodeMCU

Como se muestra en la figura anterior, se conecta el módulo GPS directo a la placa NodeMCU conectando el pin GND de GY-GPS6MV2 al pin GND de NodeMCU y el pin VCC al pin 3v3. También se conectan RXD a D1 y TXD a D2.

La mayoría de los otros módulos GPS necesitan una potencia de 5v, pero Neo-6m funciona con 3.3v, por lo que podemos alimentar directamente el módulo GPS desde el módulo NodeMCU.

```

-----
23:02:56.809 -> Connecting to Fibertel WiFi395 5.8GHz
23:02:57.299 -> .....
23:03:01.038 -> WiFi connected
23:03:01.038 -> Server started
23:03:01.038 -> 192.168.0.236
    
```

Figura 47: Salida puerto serie al momento de enviar la ubicación

Del puerto serie obtuvimos la dirección IP (192.168.0.236) la cual al ingresarla en el buscador nos dirige a una web que tiene la información necesaria para saber la ubicación exacta del paciente.

GPS Interfacing with NodeMCU

Location Details

Latitude	-34.598236
Longitude	-58.429989
Date	14 / 10 / 2021
Time	07 : 44 : 21 AM

[Click here!](#) To check the location in Google maps.

Figura 48: Interfaz de módulo GPS para la ubicación

Podremos buscar directamente en google maps para verificar la ubicación que nos otorga el módulo introduciendo la latitud y la longitud del módulo en el mapa, sin embargo, añadimos el botón “Click here”, el cual nos direccionara directamente a la posición del google maps del paciente.

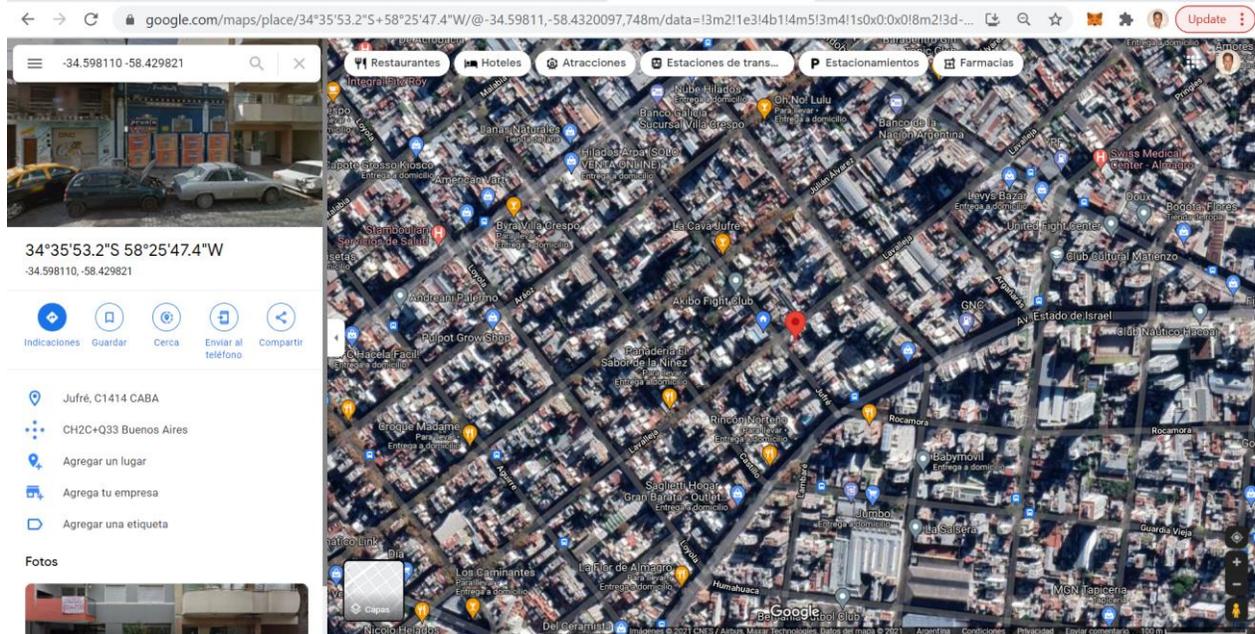


Figura 49: Ubicación del paciente prueba

4.5. Funcionamiento del MPU6050

Como habíamos comentado anteriormente, el módulo MPU6050 se usó como sensor para detectar la caída del paciente, como todos sabemos, detectar caídas es muy útil para las personas mayores en la vida diaria. Porque puede avisar a la persona o familiar cuando detecta una caída y reduce el riesgo de retraso en la atención médica. Hoy en día, podemos encontrar detectores de caídas incluso en relojes inteligentes, rastreadores de actividad física y otros tipos de wearables. Los dispositivos detectores de caídas basados en IoT pueden salvar una vida en caso de emergencia. Mediante el uso de este sensor y el NodeMCU, logramos mandar un aviso mediante la página web IFTTT antes mencionada.

Para lograr esto, usamos el giroscopio y el acelerómetro. El giroscopio se utiliza para determinar la orientación y el acelerómetro proporciona información sobre el parámetro angular, como los datos de los ejes X, Y y Z. Para detectar la caída, compararemos la magnitud de la aceleración con el valor umbral. Si se detecta la caída, el dispositivo enviará un e-mail a la

persona interesada o al médico que realiza el seguimiento. El NodeMCU ESP8266 se utiliza aquí como microcontrolador y módulo Wi-Fi para conectarse con IFTTT para enviar el mensaje.

A continuación se muestra el diagrama de conexiones para la interfaz del módulo de giroscopio y acelerómetro MPU6050 con el NodeMCU.

Conexión del módulo

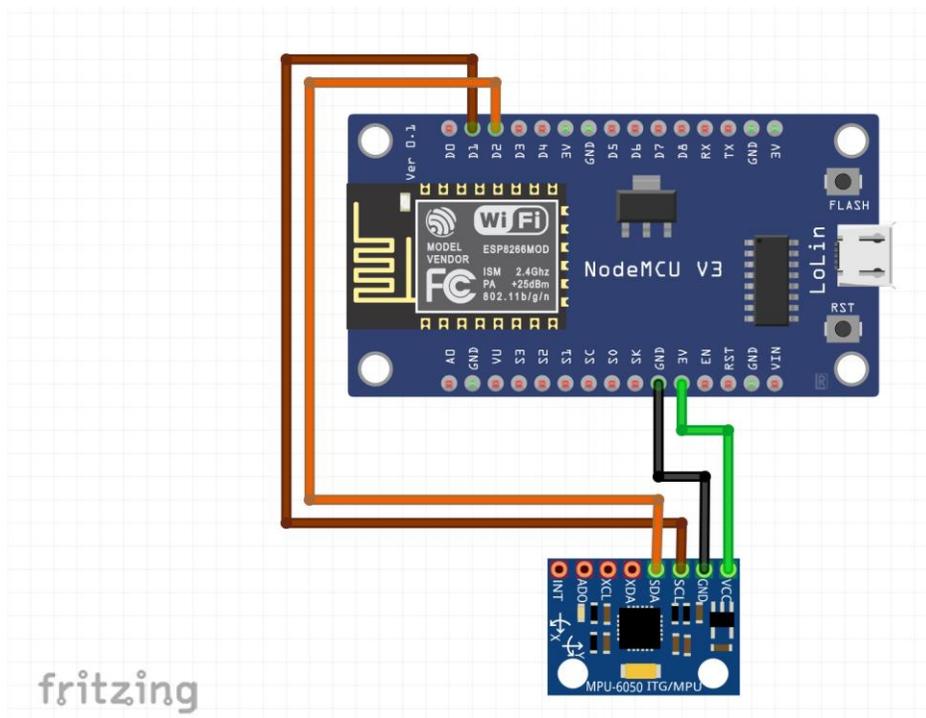


Figura 50: Esquema de conexión MPU6050 con NodeMCU

El MPU6050 se comunica con el NodeMCU a través del protocolo I2C, por lo que solo necesitamos dos cables para conectar el NodeMCU y el MPU6050. Los pines SCL y SDA de MPU6050 están conectados a los pines D22 y D21 del NodeMCU, mientras que los pines VCC y GND de MPU6050 están conectados a 3V y GND de NodeMCU.

```

COM4
19:17:56.382 -> 191
19:17:56.382 -> TRIGGER 3 ACTIVATED
19:17:56.467 -> 5
19:17:56.599 -> 9
19:17:56.683 -> 10
19:17:56.783 -> 10
19:17:56.884 -> 10
19:17:56.977 -> 10
19:17:57.106 -> 10
19:17:57.187 -> 10
19:17:57.312 -> 10
19:17:57.312 -> 5
19:17:57.312 -> 5
19:17:57.312 -> FALL DETECTED
19:17:57.312 -> Connecting to maker.ifttt.com
    
```

Figura 51: Salida del puerto serie al momento de detectar la caída del paciente

Por último, mostramos cómo llega el mensaje a la persona encargada de recibirlo en cuestión de segundos para así poder tomar las acciones pertinentes según sea el caso o la emergencia del paciente.

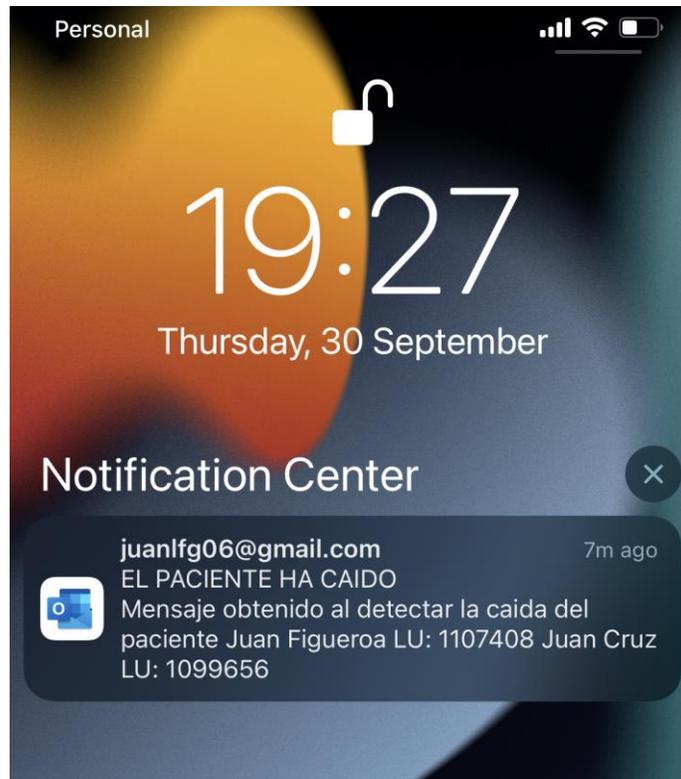


Figura 52: Notificación de la caída del paciente

5. ANÁLISIS COMERCIAL

5.1. Análisis de precios

Buscando la factibilidad y rentabilidad desde el punto de vista económico del proyecto, se realizará un análisis del precio de la construcción y armado del prototipo.

Para esto, se realizó el método en el cual basándonos en el costo de cada producto se le añadirá un margen aceptable para el valor final estimado del mismo.

Tabla I: Lista de materiales y costo del proyecto

Producto	Descripción	Precio (AR\$)
NodeMCU	Microprocesador	\$ 1,200.00
AD8232	Sensor ECG	\$ 1,600.00
MPU6050	Sensor de caída	\$ 700.00
MAX30102	Sensor Oxímetro de pulso	\$ 1,000.00
MLX90614	Sensor de temperatura	\$ 2,000.00
GY-GPS6MV2	Sensor GPS	\$ 1,500.00
	Costo Total	\$ 8,000.00

$$\text{Precio del producto} = \frac{\text{Costo}}{1 - \% \text{ margen}}$$

Considerando un margen de 30%:

$$\text{Precio del producto} = \frac{8.000 \text{ AR\$}}{1 - 0.3} = 11.428,57 \text{ AR\$}$$

Se observa que el precio del prototipo desarrollado ronda aproximadamente los 11.500 pesos argentinos, que hoy en día tomando como base el tipo de cambio 1 dólar igual a 200 pesos, tendría un valor aproximado a los 60 dólares, lo cual entra en el rango de un equipo low-cost para el área de la medicina.

5.2. Análisis de estrategias (FODA estratégico)

Para evaluar todos los factores tanto internos como externos que interfieren en el desarrollo del proyecto, se realizó un análisis FODA estratégico.

Tabla II: FODA estratégico

Amenazas	Implementación prematura de sensores suficientemente aceptables médicamente en la IoMT.
Oportunidades	El “Internet of things” es un área de la tecnología que se encuentra en auge, por lo cual se está implementando en todas las áreas posibles.
Fortalezas	Contar con el apoyo de búsqueda de nuevas soluciones para los problemas habitacionales en los establecimientos de la salud, así como la búsqueda del Estado en nuevos métodos de abordar las crisis existentes.
Debilidades	No contar con la información necesaria para cubrir todo lo que un monitoreo a distancia necesita. Así como tampoco la aceptación de la sociedad para la implementación de nuevas tecnologías en el área de la salud.

6. DISCUSIÓN

Se puede observar que el resultado de las pruebas es exitoso y nuestro dispositivo cumple con los requerimientos definidos en el alcance inicial del proyecto, permitiéndole al médico interactuar con la plataforma Ubidots y obtener los parámetros vitales del paciente que se pueden utilizar como historial médico en un futuro. La solución de IoT cumple con la condición de ser de bajo costo y con las preferencias de los usuarios de acuerdo a la encuesta realizada.

El proyecto presenta una ventaja de precio por sobre los productos de la competencia, ya que, para realizar todas las funciones mencionadas con anterioridad, diferentes marcas deben ser integradas entre sí, muchas veces de manera no simple y su instalación requiere de cierto nivel de experiencia, o de la asistencia de un experto. A su vez, en estos casos se incrementa el precio de la solución final de manera considerable.

Aunque nuestro modelo está probado e implementado, será difícil continuar el proyecto sin un soporte de hardware de calidad superior junto con una gran cantidad de nuevas integraciones. El beneficio real de este trabajo solo se puede realizar plenamente cuando se puede implementar a gran escala.

Desafíos:

Hay algunos desafíos básicos que enfrentamos durante la implementación del proyecto.

Módulo NodeMcu

En muchos casos el módulo Wi-Fi no logra conectarse con la red local, como resultado, el envío de datos se interrumpió muchas veces. Por lo tanto, se espera que un mejor soporte de hardware para el módulo Wi-Fi envíe datos sin problemas.

Sensor de pulso

Los datos que se recuperan del sensor de pulso muchas veces dan errores en las lecturas. Debido a la falta de disponibilidad de un mejor sensor de pulso en nuestro país, tuvimos que utilizar el que teníamos disponible e implementar algunos cambios en el código. Esto retrasó nuestra implementación del proyecto de alguna manera, ya que el sensor de pulso es esencial para la implementación de este proyecto.

Análisis de datos

El análisis de datos con Ubidots y el envío de notificación de caída del paciente nos dieron algunos desafíos. Especialmente con el mensaje de alarma, era difícil vincularlo con la bandeja de entrada del mail.

Los retos del IoT de la salud

- Alto coste. Si bien el retorno de la inversión es positivo y está demostrado que el IoT de la salud abarata los cuidados médicos, arrancar cada nueva infraestructura (hardware, aplicaciones, aprendizaje) sí supone un alto coste en tiempo, esfuerzo y dinero.
- Seguridad. Los datos médicos son una información extremadamente sensible que cada paciente quiere preservar. La seguridad formará parte de la calidad del producto. Si hay riesgo de violación de datos, el sistema no podrá considerarse un éxito por buenos que sean los resultados médicos.
- Estandarización. Existen ya multitud de dispositivos y programas. Pero se necesita un estándar para que puedan trabajar juntos sin interferencias y se garantice la estabilidad del sistema.
- Reglamentación. En cualquier país del mundo, las leyes respecto a medicina y salud están en los principios del sistema legal. Con la misma seriedad debe acometer la legislación del IoMT.

Trabajos futuros

- Integrar más sensores para una adquisición y análisis de datos más específicos.
- Utilizar nuestra solución para brindar servicio de salud a zonas rurales a precio asequible.
- Crear una base de datos más grande para que los médicos diagnostiquen a personas de diferentes áreas y culturas.

Nuestro proyecto puede considerarse como una plataforma a desarrollar en el campo del IoT en el sector salud. En países en desarrollo como el nuestro, este tipo de proyecto innovador y rentable puede mejorar el futuro de la tecnología. Por lo tanto, esperamos implementar el proyecto para tener un impacto en la nueva era de la tecnología.

GLOSARIO

Bluetooth: Especificación tecnológica para redes inalámbricas que permite la transmisión de voz y datos entre distintos dispositivos mediante una radiofrecuencia segura (2,4 GHz).

Hardware: Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

IR: Sigla correspondiente a Infra-Rojo, haciendo referencia a las señales pertenecientes a ese espectro.

Protoboard: Placa que posee unos orificios conectados eléctricamente entre sí siguiendo un patrón horizontal o vertical. Es empleada para realizar pruebas de circuitos electrónicos, insertando en ella componentes electrónicos y cables como puente.

Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

Tablet: Dispositivo electrónico portátil de pantalla táctil que cuenta con un tamaño intermedio entre una computadora y un celular.

Wi-Fi: Tecnología que permite la interconexión inalámbrica de dispositivos electrónicos. Los dispositivos habilitados con Wi-Fi pueden conectarse entre sí o a internet a través de un punto de acceso de red inalámbrica.

BIBLIOGRAFÍA

1. Mitma, René. Internet de las cosas médicas (IoMT), tecnología aplicada a la salud. [En línea] Junio 2021. <https://www.ambit-bst.com/blog/internet-de-las-cosas-m%C3%A9dicas-iomt-tecnolog%C3%ADa-aplicada-a-la-salud>
2. Kiversal, Qué es la IoMT. [En línea] Mayo 2018. <https://blog.kiversal.com/que-es-la-iomt/>
3. Kiversal, El diseño ergonómico de Audixi. [En línea] Julio 2017. <https://blog.kiversal.com/disenio-ergonomico-audixi-10/>
4. JUÁREZ, Benito. FIFARMA. [En línea] Febrero 2020. <https://fifarma.org/es/iomt/>
5. CASCALLANA, Tania Alonso. IOMT: la importancia del internet de las cosas médicas en contextos como el actual. [En línea] Marzo 2020. <https://hablemosdeempresas.com/empresa/iomt/>
6. MARTINEZ, Noelia. Pastillas inteligentes, audiómetros digitales y plataformas médicas, así es ya el IoT de la salud. [En línea] Agosto 2019. https://blog.orange.es/innovacion/pastillas-inteligentes-audiometros-digitales-y-plataformas-medicas-asi-es-ya-el-iot-de-la-salud/?_ga=2.171792864.1576203006.1631564540-678926839.1630713588
7. Long Nguyen, Robert Perry, Lou Seneres, Neda Seyedmahmoud. Analog Integrated Circuit Applications. [En línea]. https://web.wpi.edu/Pubs/E-project/Available/E-project-043015-041512/unrestricted/mqp_report_combined_4_28_15.pdf
8. HERNANDEZ, Maria. Connect a NodeMCU ESP8266 to Ubidots over HTTP. [En línea]. Diciembre 2021. <https://help.ubidots.com/en/articles/513312-connect-a-nodemcu-esp8266-to-ubidots-over-http>
9. AHMED, Rahma. Send multivariables to ubidots with the nodemcu esp8266?. [En línea]. Octubre 2021. <https://stackoverflow.com/questions/69483265/send-multivariables-to-ubidots-with-the-nodemcu-esp8266>

10. ELSEIVER. Escala de Coma de Glasgow: tipos de respuesta motora y su puntuación. [En línea]. Junio 2017. <https://www.elsevier.com/es-es/connect/medicina/escala-de-coma-de-glasgow>
11. LASKOWSKI, Edward. ¿Cuál es la frecuencia cardíaca normal en reposo?-. [En línea]. Octubre 2020. <https://www.mayoclinic.org/es-es/healthy-lifestyle/fitness/expert-answers/heart-rate/faq-20057979>
12. CAVACO, Juana. Niveles bajos y normales de oxígeno en la sangre: Qué necesitas saber. [En línea]. Agosto 2020. <https://www.medicalnewstoday.com/articles/es/saturacion-de-oxigeno-normal#niveles-normales-y-bajos>

ANEXO A

Código programado en lenguaje compatible con Arduino, que será cargado en el hardware NodeMCU para el funcionamiento del sensor de caída MPU6050

```

1.  #include <Wire.h>
2.  #include <ESP8266WiFi.h>
3.  const int MPU_addr=0x68; // I2C Direccion I2C del MPU-6050
4.  int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
5.  float ax=0, ay=0, az=0, gx=0, gy=0, gz=0;
6.  boolean fall = false; //Guarda el dato si ha caido el paciente
7.  boolean trigger1=false; //Guarda el dato si se ha producido el primer disparo (umbral inferior)
8.  boolean trigger2=false; //Guarda el dato si se ha producido un segundo disparo (umbral superior)
9.  boolean trigger3=false; //Guarda el dato si se ha producido el tercer disparador (cambio de orientación)
10. byte trigger1count=0; //Guarda el conteo pasado desde que el activador 1 se estableció en verdadero
11. byte trigger2count=0; //Guarda el conteo pasado desde que el activador 2 se estableció en verdadero
12. byte trigger3count=0; //Guarda el conteo pasado desde que el activador 3 se estableció en verdadero
13. int angleChange=0;
14. // Informacion del WiFi
15. const char *ssid = "Fibertel WiFi395 5.8GHz"; // Nombre del WiFi
16. const char *pass = "noporfavor1"; // Contraseña del WiFi
17. void send_event(const char *event);
18. const char *host = "maker.ifttt.com";
19. const char *privateKey = "bOfxNXR71qGnxnJHOuLrD3";

```

```

20. void setup(){
21.   Serial.begin(115200);
22.   Wire.begin();
23.   Wire.beginTransmission(MPU_addr);
24.   Wire.write(0x6B); // Registro PWR_MGMT_1
25.   Wire.write(0); // Setea en 0 el MPU-6050
26.   Wire.endTransmission(true);
27.   Serial.println("Wrote to IMU");
28.   Serial.println("Connecting to ");
29.   Serial.println(ssid);
30.   WiFi.begin(ssid, pass);
31.   while (WiFi.status() != WL_CONNECTED)
32.     {
33.       delay(500);
34.       Serial.print("."); // Imprime ... hasta que conecte al internet
35.     }
36.   Serial.println("");
37.   Serial.println("WiFi connected");
38.   }
39. void loop(){
40.   mpu_read();
41.   ax = (AcX-7)/16384.00;
42.   ay = (AcY-3)/16384.00;
43.   az = (AcZ+6)/16384.00;
44.   gx = (GyX)/131.07;
45.   gy = (GyY)/131.07;
46.   gz = (GyZ)/131.07;
47.   // Calculo de la amplitud de los vectores para los 3 ejes
48.   float Raw_Amp = pow(pow(ax,2)+pow(ay,2)+pow(az,2),0.5);

```

```

49.   int Amp = Raw_Amp * 10; // Multiplica por 10 los valores bcz cuandoe esten entre 0 to 1
50.   Serial.println(Amp);
51.   if (Amp<=2 && trigger2==false){ // Si el AM supera el valor de umbral (0.4g)
52.       trigger1=true;
53.       Serial.println("TRIGGER 1 ACTIVATED");
54.   }
55.   if (trigger1==true){
56.       trigger1count++;
57.       if (Amp>=12){ //si el AM supera el valor de umbral (3g)
58.           trigger2=true;
59.           Serial.println("TRIGGER 2 ACTIVATED");
60.           trigger1=false; trigger1count=0;
61.       }
62.   }
63.   if (trigger2==true){
64.       trigger2count++;
65.       angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5); Serial.println(angleChange);
66.       if (angleChange>=30 && angleChange<=400){ //Si la orientacion cambia entre 80-100
        grados
67.           trigger3=true; trigger2=false; trigger2count=0;
68.           Serial.println(angleChange);
69.           Serial.println("TRIGGER 3 ACTIVATED");
70.       }
71.   }
72.   if (trigger3==true){
73.       trigger3count++;
74.       if (trigger3count>=10){
75.           angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
76.           //delay(10);

```

```

77.     Serial.println(angleChange);
78.     if ((angleChange>=0) && (angleChange<=10)){ //Si la orientacion cambia entre 0-10
        grados
79.         fall=true; trigger3=false; trigger3count=0;
80.         Serial.println(angleChange);
81.         }
82.     else{ //Orientacion original del usuario
83.         trigger3=false; trigger3count=0;
84.         Serial.println("TRIGGER 3 DEACTIVATED");
85.     }
86. }
87. }
88. if (fall==true){ //Cuando se detecta la caida
89.     Serial.println("FALL DETECTED");
90.     send_event("fall_detect");
91.     fall=false;
92. }
93. if (trigger2count>=6){ //acepta 0.5s para que cambie la orientacion
94.     trigger2=false; trigger2count=0;
95.     Serial.println("TRIGGER 2 DEACTIVATED");
96. }
97. if (trigger1count>=6){ //acepta 0.5s para que el AM rompa el umbral superior
98.     trigger1=false; trigger1count=0;
99.     Serial.println("TRIGGER 1 DEACTIVATED");
100. }
101. delay(100);
102. }
103. void mpu_read(){
104.     Wire.beginTransmission(MPU_addr);

```

```

105.   Wire.write(0x3B); // comienza a registrar en la direccion 0x3B (ACCEL_XOUT_H)
106.   Wire.endTransmission(false);
107.   Wire.requestFrom(MPU_addr,14,true); // Solicita un total de 14 registros
108.   AcX=Wire.read()<<8|Wire.read();      // 0x3B  (ACCEL_XOUT_H)  &  0x3C
      (ACCEL_XOUT_L)
109.   AcY=Wire.read()<<8|Wire.read();      // 0x3D  (ACCEL_YOUT_H)  &  0x3E
      (ACCEL_YOUT_L)
110.   AcZ=Wire.read()<<8|Wire.read();      // 0x3F  (ACCEL_ZOUT_H)  &  0x40
      (ACCEL_ZOUT_L)
111.   Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
112.   GyX=Wire.read()<<8|Wire.read();      // 0x43  (GYRO_XOUT_H)  &  0x44
      (GYRO_XOUT_L)
113.   GyY=Wire.read()<<8|Wire.read();      // 0x45  (GYRO_YOUT_H)  &  0x46
      (GYRO_YOUT_L)
114.   GyZ=Wire.read()<<8|Wire.read();      // 0x47  (GYRO_ZOUT_H)  &  0x48
      (GYRO_ZOUT_L)
115.   }
116.   void send_event(const char *event)
117.   {
118.     Serial.print("Connecting to ");
119.     Serial.println(host);
120.     // Utiliza la clase WiFiClient para crear conexiones TCP
121.     WiFiClient client;
122.     const int httpPort = 80;
123.     if (!client.connect(host, httpPort)) {
124.       Serial.println("Connection failed");
125.       return;
126.     }
127.     // Se crea una URI para la solicitud

```

```

128.   String url = "/trigger/";
129.   url += event;
130.   url += "/with/key/";
131.   url += privateKey;
132.   Serial.print("Requesting URL: ");
133.   Serial.println(url);
134.   // envia la solicitud al servidor
135.   client.print(String("GET ") + url + " HTTP/1.1\r\n" +
136.               "Host: " + host + "\r\n" +
137.               "Connection: close\r\n\r\n");
138.   while(client.connected())
139.   {
140.     if(client.available())
141.     {
142.       String line = client.readStringUntil('\r');
143.       Serial.print(line);
144.     } else {
145.       // No data yet, wait a bit
146.       delay(50);
147.     };
148.   }
149.   Serial.println();
150.   Serial.println("closing connection");
151.   client.stop();
152. }

```

ANEXO B

Código programado en lenguaje compatible con Arduino, que será cargado en el hardware NodeMCU para el funcionamiento del sensor GY-GPS6MV2 utilizado como módulo GPS

```

1.  #include <TinyGPS++.h>
2.  #include <SoftwareSerial.h>
3.  #include <ESP8266WiFi.h>
4.  TinyGPSPlus gps;
5.  SoftwareSerial ss(4, 5);
6.  const char* ssid = "shashi";
7.  const char* password = "12345678";
8.  float latitude , longitude;
9.  int year , month , date, hour , minute , second;
10. String date_str , time_str , lat_str , lng_str;
11. int pm;
12. WiFiServer server(80);
13. void setup()
14. {
15.   Serial.begin(115200);
16.   ss.begin(9600);
17.   Serial.println();
18.   Serial.print("Connecting to ");
19.   Serial.println(ssid);
20.   WiFi.begin(ssid, password);
21.   while (WiFi.status() != WL_CONNECTED)
22.   {
23.     delay(500);
24.     Serial.print(".");
25.   }

```

```

26. Serial.println("");
27. Serial.println("WiFi connected");
28. server.begin();
29. Serial.println("Server started");
30. // Print the IP address
31. Serial.println(WiFi.localIP());
32. }
33. void loop()
34. {
35.   while (ss.available() > 0)
36.     if (gps.encode(ss.read()))
37.     {
38.       if (gps.location.isValid())
39.       {
40.         latitude = gps.location.lat();
41.         lat_str = String(latitude , 6);
42.         longitude = gps.location.lng();
43.         lng_str = String(longitude , 6);
44.       }
45.       if (gps.date.isValid())
46.       {
47.         date_str = "";
48.         date = gps.date.day();
49.         month = gps.date.month();
50.         year = gps.date.year();
51.         if (date < 10)
52.           date_str = '0';
53.         date_str += String(date);
54.         date_str += " / ";

```

```

55.     if (month < 10)
56.         date_str += '0';
57.     date_str += String(month);
58.     date_str += " / ";
59.     if (year < 10)
60.         date_str += '0';
61.     date_str += String(year);
62.     }
63.     if (gps.time.isValid())
64.     {
65.         time_str = "";
66.         hour = gps.time.hour();
67.         minute = gps.time.minute();
68.         second = gps.time.second();
69.
70.         minute = (minute + 30);
71.         if (minute > 59)
72.         {
73.             minute = minute - 60;
74.             hour = hour + 1;
75.         }
76.         hour = (hour + 5) ;
77.         if (hour > 23)
78.             hour = hour - 24;
79.         if (hour >= 12)
80.             pm = 1;
81.         else
82.             pm = 0;
83.         hour = hour % 12;

```

```

84.     if (hour < 10)
85.         time_str = '0';
86.     time_str += String(hour);
87.     time_str += " : ";
88.     if (minute < 10)
89.         time_str += '0';
90.     time_str += String(minute);
91.     time_str += " : ";
92.     if (second < 10)
93.         time_str += '0';
94.     time_str += String(second);
95.     if (pm == 1)
96.         time_str += " PM ";
97.     else
98.         time_str += " AM ";
99.
100.    }
101.    }
102.    // Check if a client has connected
103.    WiFiClient client = server.available();
104.    if (!client)
105.    {
106.        return;
107.    }
108.    // Prepare the response
109.    String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n <!DOCTYPE html>
<html> <head> <title>GPS Interfacing with NodeMCU</title> <style>";
110.    s += "a:link {background-color: YELLOW;text-decoration: none;}";
111.    s += "table, th, td {border: 1px solid black;} </style> </head> <body> <h1 style=";

```

```

112.     s += "font-size:300%;";
113.     s += " ALIGN=CENTER> GPS Interfacing with NodeMCU</h1>";
114.     s += "<p ALIGN=CENTER style='\"font-size:150%;\"'";
115.     s += "> <b>Location Details</b></p> <table ALIGN=CENTER style=";
116.     s += "width:50%";
117.     s += "> <tr> <th>Latitude</th>";
118.     s += "<td ALIGN=CENTER >";
119.     s += lat_str;
120.     s += "</td> </tr> <tr> <th>Longitude</th> <td ALIGN=CENTER >";
121.     s += lng_str;
122.     s += "</td> </tr> <tr> <th>Date</th> <td ALIGN=CENTER >";
123.     s += date_str;
124.     s += "</td></tr> <tr> <th>Time</th> <td ALIGN=CENTER >";
125.     s += time_str;
126.     s += "</td> </tr> </table> ";
127.     if (gps.location.isValid())
128.     {
129.         s += "<p align=center><a style='\"color:RED;font-size:125%;\"";
130.         s += lat_str;
131.         s += "+";
132.         s += lng_str;
133.         s += "\"\" target='\"_top\"\">Click here!</a> To check the location in Google maps.</p>";
134.     }
135.     s += "</body> </html> \n";
136.     client.print(s);
137.     delay(100)
138. }

```

ANEXO C

Código programado en lenguaje compatible con Arduino, que será cargado en el hardware NodeMCU para el funcionamiento del sensor MLX90614 y envío de datos a UBIDOTS.

```

1.  #include "Ubidots.h"
2.  #include <Wire.h>
3.  #include <Adafruit_MLX90614.h>
4.  Adafruit_MLX90614 mlx = Adafruit_MLX90614();
5.  const char* UBIDOTS_TOKEN = "BBFF-VhSDeMqRRmwzSbvNDIttryu0SXFPN"; //
    Put here your Ubidots TOKEN
6.  const char* WIFI_SSID = "Fibertel WiFi754 2.4GHz"; // Put here your Wi-Fi SSID
7.  const char* WIFI_PASS = "0019579418"; // Put here your Wi-Fi password
8.  Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);
9.  void setup() {
10.   Serial.begin(9600);
11.   ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
12.   // ubidots.setDebug(true);
13.   Serial.println("Adafruit test");
14.   mlx.begin();
15. }
16. void loop() {
17.   Serial.print("*C\tCuerpo = "); Serial.print(mlx.readObjectTempC()); Serial.println("*C");
18.   Serial.println();
19.   float value1 = mlx.readObjectTempC();
20.   ubidots.add("Variable_Name_One", value1); // Envío la variable temperatura del cuerpo
21.   bool bufferSent = false;
22.   bufferSent = ubidots.send();

```

```

23.   if (bufferSent) {
24.     // Do something if values were sent properly
25.     Serial.println("Values sent by the device");
26.   }
27.   Serial.print("*C\tCuerpo = "); Serial.print(mlx.readObjectTempC()); Serial.println("*C");
28.   Serial.println();
29.   delay(5000);
30. }

```

ANEXO D

Código programado en lenguaje compatible con Arduino, que será cargado en el hardware NodeMCU para el funcionamiento del sensor AD8232 y envío de datos a UBIDOTS.

```

1.   #include <PubSubClient.h>
2.   #include <ESP8266WiFi.h>
3.   #define WIFISSID "Fibertel WiFi513 2.4GHz"
4.   #define PASSWORD "01033633131819"
5.   #define TOKEN "BBFF-VhSDeMqRRmwzSbvNDItytryu0SXFPN"
6.   #define MQTT_CLIENT_NAME "myecgsensor"
7.   #define VARIABLE_LABEL "myecg"
8.   #define DEVICE_LABEL "esp8266"
9.   #define SENSOR A0
10.  char mqttBroker[] = "industrial.api.ubidots.com";
11.  char payload[100];
12.  char topic[150];
13.  char str_sensor[10];
14.  WiFiClient ubidots;

```

```

15. PubSubClient client(ubidots);
16. void callback(char* topic, byte* payload, unsigned int length) {
17.     char p[length + 1];
18.     memcpy(p, payload, length);
19.     p[length] = NULL;
20.     Serial.write(payload, length);
21.     Serial.println(topic);
22. }
23. void reconnect() {
24.     while (!client.connected()) {
25.         Serial.println("Attempting MQTT connection...");
26.         if (client.connect(MQTT_CLIENT_NAME, TOKEN, "")) {
27.             Serial.println("Connected");
28.         } else {
29.             Serial.print("Failed, rc=");
30.             Serial.print(client.state());
31.             Serial.println(" try again in 2 seconds");
32.             delay(2000);
33.         }
34.     }
35. }
36. void setup() {
37.     Serial.begin(115200);
38.     WiFi.begin(WIFISSID, PASSWORD);
39.     pinMode(SENSOR, INPUT);
40.     Serial.println();
41.     Serial.print("Waiting for WiFi...");
42.     while (WiFi.status() != WL_CONNECTED) {
43.         Serial.print(".");

```

```

44.     delay(500);
45.   }
46.   Serial.println("");
47.   Serial.println("WiFi Connected");
48.   Serial.println("IP address: ");
49.   Serial.println(WiFi.localIP());
50.   client.setServer(mqttBroker, 1883);
51.   client.setCallback(callback);
52. }
53. void loop() {
54.   if (!client.connected()) {
55.     reconnect();
56.   }
57.   sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
58.   sprintf(payload, "%s", "");
59.   sprintf(payload, "{\"%s\":", VARIABLE_LABEL);
60.   float myecg = analogRead(SENSOR);
61.   dtostrf(myecg, 4, 2, str_sensor);
62.   sprintf(payload, "%s {\"value\": %s}", payload, str_sensor);
63.   Serial.println("Publishing data to Ubidots Cloud");
64.   client.publish(topic, payload);
65.   client.loop();
66.   delay(10);
67. }

```

ANEXO E

Código programado en lenguaje compatible con Arduino, que será cargado en el hardware NodeMCU para el funcionamiento del sensor MAX30100 y envío de datos a UBIDOTS.

```

1.  #include "Ubidots.h"
2.  #include <ESP8266WiFi.h>
3.  #include <Wire.h>
4.  #include "MAX30105.h"
5.  #include "spo2_algorithm.h"
6.  const char* UBIDOTS_TOKEN = "BBFF-VhSDeMqRRmwzSbvNDIttryu0SXFPN";
7.  const char* WIFI_SSID = "Fibertel WiFi513 2.4GHz";
8.  const char* WIFI_PASS = "01033633131819";
9.  #define output A0
10. const int N=10;
11. int ecg[N];
12. int n= 0;
13. float ecgmoy =0 ;
14. float somme =0;
15. MAX30105 particleSensor;
16. uint32_t irBuffer[100];
17. uint32_t redBuffer[100];
18. int32_t bufferLength;
19. int32_t spo2;
20. int8_t validSPO2;
21. int32_t heartRate;
22. int8_t validHeartRate;
23. char mqttBroker[] = "industrial.api.ubidots.com";
24. char payload[100];
25. char topic[150];

```

```

26.  char str_val_1[10];
27.  char str_val_2[10];
28.  char str_val_3[10];
29.  Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);
30.  void setup() {
31.      Serial.begin(115200);
32.      ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
33.      pinMode(output, INPUT);
34.      pinMode (LED_BUILTIN, OUTPUT);
35.      Serial.print("Inicializando Oximetro de pulso...");
36.      if (!particleSensor.begin(Wire, I2C_SPEED_FAST))
37.      {
38.          Serial.println(F("MAX30102 mal inicializado"));
39.          while (1);
40.      }
41.      Serial.println(F("Colocar el dedo sobre el sensor"));
42.      while (Serial.available() == 0) ;
43.      Serial.read();
44.      byte ledBrightness = 60;
45.      byte sampleAverage = 4;
46.      byte ledMode = 2;
47.      byte sampleRate = 100;
48.      int pulseWidth = 411;
49.      int adcRange = 4096;
50.      particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth,
adcRange);
51.      for (int i=0 ; i<N ; i++)
52.      {
53.          ecg [i] = 0;

```

```

54.     }
55.   }
56.   void loop() {
57.     Serial.begin (115200);
58.     bufferSize = 100;
59.     for (byte i = 0 ; i < bufferSize ; i++)
60.     {
61.       while (particleSensor.available() == false)
62.         particleSensor.check();
63.       redBuffer[i] = particleSensor.getRed();
64.       irBuffer[i] = particleSensor.getIR();
65.       particleSensor.nextSample();
66.     }
67.     maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferSize, redBuffer, &spo2,
&validSPO2, &heartRate, &validHeartRate);
68.     while (1) {
69.       for (byte i = 25; i < 100; i++) { redBuffer[i - 25] = redBuffer[i];
70.                                     irBuffer[i - 25] = irBuffer[i]; }
71.       for (byte i = 75; i < 100; i++)
72.       {
73.         while (particleSensor.available() == false)
74.           particleSensor.check();
75.         redBuffer[i] = particleSensor.getRed();
76.         irBuffer[i] = particleSensor.getIR();
77.         particleSensor.nextSample();
78.         Serial.print(F(", HR="));
79.         Serial.print(heartRate, DEC);
80.         Serial.print(F(", SPO2="));
81.         Serial.print(spo2, DEC);

```

```

82.     }
83.     float value1 = heartRate;
84.     float value2 = spo2;
85.     ubidots.add("Pulso", value1); // Envio la variable de pulso
86.     ubidots.add("% Oxigeno en la sangre", value2); // Envio la variable de oxigeno en sangre
87.     bool bufferSent = false;
88.     bufferSent = ubidots.send();
89.     if (bufferSent) {
90.         Serial.println("Valores enviados");
91.     }
92.     maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2,
        &validSPO2, &heartRate, &validHeartRate);
93.     }
94. }
95.

```