UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE MÓDULO DIDÁCTICO IOT APLICADO A TELEMEDICINA CON HARDWARE DE BAJO COSTO Y SERVICIOS WEB EN LA NUBE

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

AUTORES: CRISTHIAN AGUSTÍN MACÍAS ZAMBRANO
MIGUEL ÁNGEL ORTÍZ VILLEGAS
TUTOR: ING. VÍCTOR DAVID LARCO TORRES MSC.

Guayaquil – Ecuador

2021

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Cristhian Agustín Macías Zambrano con documento de identificación N° 0927565408 y Miguel Ángel Ortiz Villegas con documento de identificación N° 0925103251; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 11 de octubre del año 2021

Atentamente,

Cristhian Agustín Macías Zambrano
0927565408

Miguel Ángel Ortiz Villegas
0925103251

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Cristhian Agustín Macías Zambrano con documento de identificación N° 0927565408 y Miguel Ángel Ortiz Villegas con documento de identificación N° 0925103251, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del proyecto de investigación: “Diseño e implementación de módulo didáctico IOT aplicado a telemedicina con hardware de bajo costo y servicios web en la nube ”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 11 de octubre del año 2021

Atentamente,

Cristhian Agustín Macías Zambrano
0927565408

Miguel Ángel Ortiz Villegas
0925103251

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Víctor David Larco Torres con documento de identificación N° 0923270136, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE MÓDULO DIDÁCTICO IOT APLICADO A TELEMEDICINA CON HARDWARE DE BAJO COSTO Y SERVICIOS WEB EN LA NUBE , realizado por Cristhian Agustín Macías Zambrano con documento de identificación N° 0927565408 y Miguel Ángel Ortiz Villegas con documento de identificación N° 0925103251, obteniendo como resultado final el trabajo de titulación bajo la opción tesis que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 11 de octubre del año 2021

Atentamente,

Ing. Víctor David Larco Torres, MSc
0923270136

DEDICATORIA

CRISTHIAN AGUSTÍN MACÍAS ZAMBRANO

En primer lugar, a Jehová por haberme brindado el don de la sabiduría para poder escoger el camino indicado.

Mis padres mis pilares fundamentales, los que siempre han estado ahí a pesar de las adversidades y dificultades que se me han presentado en mi vida, gracias a sus valores y principios que me han inculcado y la confianza depositada han logrado formar un hombre de bien.

Mi hermana digna de mi admiración por las cosas que se propone lucha por conseguirlas.

A mi abuelita que se encuentra en el cielo, la que siempre me supo llenar con un consejo para realizar las cosas de la mejor manera.

A mi abuelita Barbara Santos, la que siempre me decía que Dios le preste vida para poder estar en mis logros alcanzados.

A mi familiares y amigos que siempre estuvieron ahí acompañándome en este proceso.

Mis profesores que con su ardua dedicación de enseñanza me guiaron de la manera correcta en el proceso de obtener el título universitario especialmente a mi tutor por la paciencia en impartirme sus conocimientos.

Finalmente, a Kathiuska Yovani por tu apoyo y paciencia incondicional, ser ese complemento en mi vida para seguir adelante cada día y no darme por vencido. Has sido parte de mis derrotas, tristezas, alegrías y le doy gracias a Dios por estar en mis logros.

MIGUEL ÁNGEL ORTÍZ VILLEGAS

Agradeciéndole infinitamente a DIOS por guiar mis pasos les dedico mi trabajo de titulación a las personas que con su amor lograron hacerme un hombre de bien:

Mis padres, porque a lo largo de mi vida siempre estuvieron dándome su apoyo incondicional por fuerte que fueran las adversidades nunca dejaron de estar a mi lado llenándome de su amor, sus sabios consejos están en mi mente este es el regalo que les puedo dar a tanto esfuerzo y dedicación hacia mí.

Mi esposa, llegaste en el momento oportuno en mi vida de tu mano me enseñaste a madurar y luchar por lo que uno quiere, cambiaste mi perspectiva de visualizar las cosas tu amor incondicional es el apoyo más grande para mis momentos difíciles este triunfo es de nosotros,

sin ti esto no fuera realidad

Mi hermana, siempre te admirado y eres mi mayor referente para realizar las cosas bien

Mi hijo, eres mi motor de lucha en esta vida por ti volví a estudiar, me esforcé y quiero que siempre estés orgulloso de mi.

Resumen del proyecto

Año	Alumnos	Tutor de Proyecto de titulación	Proyecto de titulación
2021	Cristhian Agustín Macías Zambrano Miguel Ángel Ortiz Villegas	Ing. Víctor Larco MSc.	Diseño e implementación de módulo didáctico IOT aplicado a telemedicina con hardware de bajo costo y servicios web en la nube

Las tecnologías del internet de las cosas (IoT) y de servidores web en la nube son indispensables en el desarrollo de las sociedades tecnológicas del siglo XXI, las grandes ciudades están implementando este tipo de tecnologías aplicadas a diferentes industrias una de ellas es la telemedicina, para el mejoramiento de procesos en la salud, acceso remoto a larga distancia y manejo de información que permitan la toma de decisiones en tiempos oportunos y puedan reducir costos económicos, ahorrar tiempo, costos energéticos y sobre todo el salvar vidas de las personas. En este contexto se plantea un prototipo de módulo didáctico de telemedicina aplicada al internet de las cosas que junto con hardware como el ESP32, Olimex y servicios web en la nube de AWS (Amazon Web Services) y Ubidots se puede monitorizar signos vitales como el pulso cardiaco EKG, oxigenación de la sangre, temperatura corporal, peso corporal y monitorización mediante cámara web el cual servirá para el chequeo remoto de una persona a distancia alimentando una base de datos y mediante una aplicación web visualizar datos para la toma de decisiones futuras en la salud de un paciente. Para este trabajo de investigación se plantea un prototipo de telemedicina que consiste en un banco didáctico de seis prácticas que serán usados por estudiantes de la Carrera de Ingeniería Electrónica y Automatización, estudiantes de la Carrera de Biomedicina o por estudiantes de la Carrera de Ingeniería en Telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil para la complementación de su desarrollo profesional y preparación para el campo laboral dentro de la telemedicina y el internet de las cosas al aplicar herramientas de electrónica, programación y servidores web en la nube como Ubidots y AWS que permitan obtener, registrar y monitorizar los datos obtenidos por los pacientes durante las prácticas y pruebas de prototipado.

Abstract

Year	Students	Degree Project Tutor	Technical Degree Project
2021	Cristhian Agustín Macías Zambrano Miguel Ángel Ortiz Villegas	Ing. Víctor Larco MSc.	Design and Implementation of IOT Didactic Module Applied To Telemedicine With Low-Cost Hardware And web services in the cloud .

The technologies of the Internet of Things (IoT) and web servers in the cloud are indispensable in the development of technological societies of the XXI century, large cities are implementing this type of technologies applied to different industries one of them is telemedicine, for the improvement of processes in health, remote access to long distance and information management that allow decision making in opportune times and can reduce economic costs, saving time, energy costs and above all saving people's lives. In this context, a prototype of a didactic module of telemedicine applied to the Internet of Things is proposed that together with hardware such as ESP32, Olimex and web services in the AWS cloud (Amazon Web Services) and Ubidots can monitor vital signs such as the EKG heart rate, blood oxygenation, body temperature, body weight and webcam monitoring which will serve for the remote check of a person at a distance feeding a database and through a web application visualize data for making future decisions in the health of a patient. For this research work, a prototype of telemedicine is proposed, consisting of a didactic bench of six practices that will be used by students of the electronic engineering and automation career, students of the biomedicine career or by students of the telecommunications engineering career of the Salesian Polytechnic University Guayaquil headquarters for the complementation of their professional development and preparation for the labor field within telemedicine and the Internet of Things by applying electronics, programming and web server tools in the cloud such as Ubidots and AWS that allow obtaining, recording and monitoring the data obtained by patients during prototyping practices and tests.

Índice general

Índice figuras

Introducción

La Telemedicina a nivel mundial ha incrementado su uso y operación en el contexto de la pandemia COVID19, es por esto por lo que muchos países incluido el Ecuador han impulsado el uso de la Telemedicina. Las personas se han visto en la necesidad de utilizar formalmente la Telemedicina por las circunstancias actuales de la pandemia y la obligatoriedad de mantener el distanciamiento físico. El mismo Ministerio de Salud Pública (MSP) se ha apoyado en la Telemedicina para hacer el triraje de los pacientes sospechosos de CoVID19 y, a su vez, no dejar a otras patologías crónicas sin atención médica. Asimismo, los médicos del sector privado han visto como una alternativa adecuada para seguir brindando atención a sus pacientes (Edicionmedica.ec, 2021).

De acuerdo con proyectos anteriores realizados en diferentes países se puede observar que la Telemedicina va de la mano con el internet de las cosas, como en el caso del prototipo HEMAN de Chanchal Raj, donde se explica que el sistema de salud de Telemedicina proporciona la provisión de tratamiento médico a distancia. La investigación y el desarrollo de productos de Telemedicina ha experimentado un crecimiento prodigioso durante la última década, principalmente debido al tremendo avance tecnológico en las TIC y la automatización. El objetivo del sistema de atención médica basado en IoT es garantizar y aumentar el bienestar de los pacientes y la calidad de vida en las zonas rurales. Por tal motivo se presentauna plataforma de sensores de salud de bajo costo para el monitoreo de la salud rural con una interfaz bien estructurada y segura entre expertos médicos y centros remotos para compartir parámetros médicos importantes (Raj, Jain, & Arif, 2018).

Las empresas del área de la electrónica médica hoy en día requieren de ingenieros con conocimientos prácticos en este campo de la Telemedicina por tal motivo es indispensable que los estudiantes de la Carrera de Electrónica de la Universidad Politécnica Salesiana tengan en sus laboratorios un prototipo de Telemedicina con el cual se pueda desarrollar prácticas que permita aplicar todos sus conocimientos teóricos de la Telemedicina mediante un prototipo para tal fin.

En ese contexto se propone el diseño e implementación de módulo didáctico para aplicaciones de Telemedicina con servicios web en la nube e internet de las cosas que sirva como objeto de práctica y experimentación a los futuros ingenieros en el campo de la

electrónica de la Universidad Politécnica Salesiana, con la finalidad de realizar prácticas de Telemedicina mediante el banco de pruebas propuesto.

Este trabajo de investigación se clasifica de la siguiente manera, en la primera parte se explica el problema, antecedentes, importancia y alcance, delimitación, beneficiarios de la propuesta, propuesta de solución y objetivos. En la segunda parte se refiere a los fundamentos teóricos de Telemedicina y de los sensores a utilizarse en el banco de pruebas, en la tercera parte se refiere al marco metodológico, tipo de investigación. Y en la cuarta y última parte se presentan el análisis de resultados, conclusiones y recomendaciones. Finalmente se agregan los anexos que corresponde a las prácticas.

1. El problema

1.1. Descripción del problema

En la actualidad la Universidad Politécnica Salesiana sede Guayaquil existe poca infraestructura tecnológica respecto a módulos didácticos para aplicaciones en Telemedicina. La Carrera de Ingeniería en Electrónico y Automatización actualmente posee dentro de la malla curricular el itinerario de la materia Biomédica, por lo cual se propone implementar módulos de Telemedicina para prácticas afines al itinerario.

En este contexto, se plantea el diseño e implementación de un módulo didáctico aplicado a la Telemedicina utilizando servicios web en la nube y el internet de las cosas, que permita poner en práctica conocimientos de electrónica y telecomunicaciones a los futuros ingenieros en electrónica y automatización de la Universidad Politécnica Salesiana sede Guayaquil, experimentando el uso de sistemas del internet de las cosas y los servicios web basados en la nube mediante AWS (Amazon Web Services) de la plataforma Amazon y Ubidots.

1.2. Antecedentes

La Universidad Politécnica Salesiana sede Guayaquil se ha caracterizado por ser una universidad que desarrolla investigación en el área técnica específicamente en el área de la electrónica se han desarrollado algunos prototipos y proyectos de investigación a beneficio de la sociedad en común, esto se puede corroborar revisando el dspace de la Universidad donde se encuentra la bibliografía de tesistas que han desarrollado proyectos innovadores en la Carrera de Electrónica.

Uno de los desafíos en el área de investigación en la Telemedicina es el desarrollo de aplicativos, elementos o prototipos en beneficio de la sociedad en común. Es por este motivo que se propone el Diseño e Implementación de Módulo Didáctico para Aplicaciones de Telemedicina con Servicios web en la nube e internet de las cosas.

1.3. Importancia y alcance

Con el aporte de esta propuesta, la sociedad en general se verá beneficiada al contar con investigaciones y prototipos aplicados a Telemedicina realizado por estudiantes e investigadores de la Universidad Politécnica Salesiana sede Guayaquil, de igual manera se verá beneficiados los estudiantes de la Carrera de Ingeniería en Electrónica y de Telecomunicaciones los cuales podrán poner en práctica los conocimientos adquiridos en las diferentes materias aplicando la electrónica, internet de las cosas y la Telemedicina.

1.4. Delimitación

1.4.1. Delimitación temporal

El tiempo estimado para el diseño, implementación y pruebas de funcionamiento del banco de pruebas es de 6 meses.

1.4.2. Delimitación espacial

Debido a la emergencia sanitaria por motivo de la pandemia del COVID-19 y siguiendo las indicaciones de la dirección de Carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana, sede Guayaquil, las pruebas del prototipo IoT de Telemedicina aplicado a módulo didáctico de electrocardiograma con hardware de bajo costo y servicios web en la nube se realizará en las instalaciones de la Universidad, posterior a las pruebas y presentación del proyecto, se entregará el prototipo de Telemedicina al laboratorio de electrónica asignado.

El testeo y análisis de resultados se lo hará en tiempo real de acuerdo con el diseño del prototipo IoT toda esta información obtenida del EKG, del oxímetro, temperatura y peso corporales serán administrados vía web en servidores AWS (Amazon Web Services) y en servidores Ubidots.

1.4.3. Delimitación académica

Este trabajo de investigación donde el resultado es un prototipo IoT funcional y operativo será una herramienta fundamental para el desarrollo de las capacidades y destrezas de los futuros Ingenieros Electrónicos e Ingenieros de Telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil.

Con este trabajo se pone en práctica los conocimientos adquiridos en materias como Redes de Computadora, Electrónica Analógica y Digital, Comunicaciones Digitales y Medios de Transmisión.

1.5. Beneficiarios de la propuesta

Este proyecto tendrá como beneficiarios directos a los estudiantes de la Carrera de Ingeniería Electrónica en Automatización, y de la Carrera de Telecomunicaciones puesto que se realizarán manuales informativos y prácticos de módulos didácticos de pruebas de telemedicina, aplicados a prácticas de laboratorio para el desarrollo de conocimientos en IoT, servicios web en la nube y electrónica digital en general. Específicamente a estudiantes que escojan el itinerario de Biomédica.

Adicionalmente podría aportar para futuras prácticas en estudiantes de la nueva Carrera de Ingeniería en Biomedicina de la Universidad Politécnica Salesiana sede Guayaquil.

1.6. Propuesta de solución

La necesidad de contar con módulos interactivos de Telemedicina para el desarrollo práctico de los conocimientos de los futuros Ingenieros Electrónicos en Automatización, y de Telecomunicaciones específicamente aportaría en el fortalecimiento de la práctica en el itinerario de la materia Biomédica dando como beneficio a los estudiantes que escojan este itinerario en su malla curricular.

También tiene como justificativo el aprendizaje de los futuros Ingenieros de Telecomunicaciones de la Universidad Politécnica Salesiana sede Guayaquil ya que elinternet de las cosas y los servicios web basados en la nube aportarán en el correcto desempeño práctico de sus conocimientos. Con la realización de este proyecto no solo se logrará aportar en el estudio de sistemas IoT en Telemedicina también servirá como iniciativapara proyectos de investigación dentro del campo del internet de las cosas y los servicios weben la nube basados en AWS y Ubidots.

1.7. Objetivos de la investigación

1.7.1. Objetivo general

Diseñar e implementar un módulo didáctico para aplicaciones de Telemedicina con servicios web en la nube e internet de las cosas (IoT).

1.7.2. Objetivos específicos

- Diseñar un módulo didáctico de electrocardiograma EKG con Olimex A20 y Shield EKG y electrodos de 3 vías.

- Diseñar un módulo didáctico de electrocardiograma EKG con ESP32 y

AD8232 y electrodos de 3 vías.

- Configurar servicios en la nube para el almacenamiento y monitorización de la información de los EKG con Amazon Web Services.

- Realizar manual de seis prácticas didácticas del prototipo IoT de Telemedicina.

2. Fundamentos teóricos

2.1. Telemedicina

Según la Organización Mundial de la Salud, la Telemedicina se refiere a la prestación de servicios médicos donde la distancia es el factor clave, brindados por profesionales que utilizan las tecnologías de la información y la comunicación para intercambiar información efectiva para el diagnóstico, tratamiento y prevención de enfermedades, así como la investigación y la continuidad. trabajo Formación de profesionales. Desde la aparición de Internet en 1990, la posibilidad de la Telemedicina se ha ampliado y su uso también ha aumentado en varios campos de la medicina. La Telemedicina se utiliza para conectar a dos o más médicos o un médico con un paciente e incluye diferentes métodos, como la consulta remota, la orientación remota, el monitoreo remoto, la detección y la notificación remota. (Domingues et al., 2020)

En Telemedicina, las interacciones pueden ocurrir de forma sincrónica o asincrónica en 2-tuplas, triples o comités. El modo síncrono suele producirse en tiempo real mediante videollamada, teléfono o radio. Los métodos asincrónicos (almacenar y reenviar) incluyen interacciones retardadas, como correo electrónico, mensajería en línea o plataformas web,

que publican informes remotos y evalúan el examen del paciente y los datos demográficos clínicos.

Durante la nota de sincronización, pueden producirse interacciones entre los siguientes elementos:

- Pacientes y médicos.
- Un paciente, el médico tratante y el médico especialista, son responsables de orientar al médico tratante.

- Pacientes y médicos mediados por un mediador (un profesional / técnico de la salud, no un médico, que ayuda al paciente a realizar mediciones de rutina u otras medidas necesarias en el cuerpo del paciente).

- Médicos tratantes y médicos consultores, (comité).

La Telemedicina ha cambiado el paradigma de la relación médico-paciente, estableciendo un vínculo de confianza (pilar básico de la conducta médica) sin contacto físico, visual y / o auditivo.(Mesa & Iván Pérez, 2020a)

2.1.1. Medicina remota

Es posible provocar un comportamiento médico real a través de videollamadas. En la consulta de videollamada en la que otras personas interactúan con el médico tratante y el paciente, el médico tratante continúa realizando acciones médicas. Para que los pacientes no

confundan el rol de la persona involucrada, y transmitan sus inquietudes a quien le corresponda, es necesario evitar que terceros asuman el papel de traficantes nublados. Los pacientes con dificultades de comunicación pueden ser asistidos por un mediador o familiares, y tener cuidado de no convertirse en su portavoz o interferir excesivamente en el diálogo médico-paciente.

Donde quiera que se desarrolle el acto médico, su ejecutor debe actuar con profesionalismo, atendiendo a los principios de primacía del bienestar del paciente, respeto a su autonomía y justicia social.

Se requiere que el médico cultive una autodisciplina y ética suficientes que le permitan reconocer sus propias limitaciones e identificar aquellas situaciones que no reúnen las condiciones mínimas para efectuar un acto médico seguro y beneficioso.

Además, debe dominar ciertas habilidades comunicacionales que le permitan leer atentamente el lenguaje corporal y obtener toda la información necesaria, especialmente en pacientes poco dados a expresar verbalmente lo que les ocurre.

2.1.2. Responsabilidad médica a través de la Telemedicina

Independientemente del comportamiento médico que se produzca, el albacea debe actuar con profesionalismo, atenerse primero al bienestar del paciente y respetar los principios

de autonomía del paciente y justicia social. El médico debe desarrollar suficiente autodisciplina y ética que le permitan reconocer sus limitaciones y determinar aquellas condiciones que no cumplen con las condiciones mínimas para realizar conductas médicas seguras y beneficiosas. Además, debe dominar ciertas habilidades de comunicación para poder leer el lenguaje corporal con atención y obtener toda la información necesaria, especialmente para los pacientes que no están dispuestos a expresar verbalmente lo que les está sucediendo.(Mesa & Iván Pérez, 2020b)

A pesar de la normativa anterior, aún es necesario establecer un acuerdo para definir con precisión los roles y responsabilidades del personal que interactúa en el proceso de atención remota, señalar los escenarios o situaciones en las que no se recomienda el uso de la t

Telemedicina, solucionar problemas de seguridad de la red. y tome medidas preventivas. Evite la fuga de datos confidenciales.

También se deben implementar procedimientos de consentimiento informado guiados por personal especialmente capacitado, los cuales, entre otras cosas, deben indicar claramente las ventajas y desventajas de la Telemática a utilizar. Si no existen condiciones mínimas para garantizar un comportamiento médico a distancia segura, no se puede implementar y se debe optar por un enfoque cara a cara. Ejemplos:

- Cuando es imprescindible someterse a un examen físico para realizar un diagnóstico.

- Cuando es necesario transmitir noticias desafortunadas, se requiere contención emocional y / o física, o el paciente sospecha que el paciente puede lastimarse a sí mismo o a otros.

2.2. Telemedicina en el Ecuador

En las últimas décadas, las tecnologías de la información y la comunicación aplicadas a la salud han tenido un impacto positivo en el sector de la salud en todo el mundo. Con la revolución de las TIC a fines de la década de 1990, resurgió el interés en la Telemedicina, y esta innovación puede desempeñar un papel importante en el desarrollo y la transformación del sistema de salud en los próximos años.

Dado que la Telemedicina brinda soluciones a los desafíos y desafíos actuales, como la creciente demanda de servicios médicos, el envejecimiento de la población o el envejecimiento de la población, esto no es solo desde un punto de vista técnico, sino también desde un punto de vista sociocultural y económico. vínculo crítico y disruptivo que requiere la gestión de grandes cantidades de información, etc.(Seo.com.ec, 2021)

2.2.1 Asistencia en zonas remotas

La evolución de las nuevas tecnologías propicia la expansión y aplicación de los conceptos de globalización e interoperabilidad en las organizaciones sanitarias, permitiendo y potenciando organizaciones y entornos laborales que van más allá de la aplicación de la Telemedicina en el ámbito de la salud en zonas remotas.

Con este nuevo panorama, la Telemedicina ofrece la posibilidad de brindar servicios médicos en áreas remotas o falta de expertos, y permite el uso de un enfoque multidisciplinar, mejorando así el nivel de los médicos de atención primaria. Debido a la interoperabilidad del sistema y las telecomunicaciones, la Telemedicina facilita todo el proceso de enfermería como

la administración, el diagnóstico, el tratamiento o la educación para la salud.

2.2.2 Regulación legal de la Telemedicina en Ecuador

Aunque la Telemedicina ha jugado un papel especial en la actual crisis global debido al Covid-19, este no es un problema nuevo en Ecuador. De hecho, durante 2010, el Ministerio de Salud Pública (MSP) intentó impulsar el "Programa Nacional de Telemedicina / Telemedicina" para brindar servicios médicos a comunidades de difícil acceso. El proyecto se desarrollará inicialmente en la Amazonía de Ecuador y luego se ampliará a todo el territorio nacional. Lamentablemente, no se tenía avances tecnológicos en ese momento, por lo que el proyecto no se pudo llevar a cabo de manera satisfactoria en el tiempo previsto. Sin embargo, esto sentó un precedente para las cosas años después.

Otra gran limitación que propone el proyecto es la obsolescencia del ordenamiento jurídico ecuatoriano, que es un problema aún existente, por lo que la expansión de la Telemedicina en el país se ha visto muy restringida. (Saludiario.com, 2021)

Es precisamente porque, desde 1992, está vigente el "Código de Ética Médica", en el que el artículo 101 prohíbe claramente "por correspondencia o cualquier otro medio de comunicación oral o escrita, brindar consultas e instrucciones de tratamiento a las personas que haya no ha sido examinado y se desconoce el historial médico ".

En este caso, aún existe un vacío legal en cuanto al uso de este método alternativo, que puede generar confusión entre los pacientes y el personal médico, y este método alternativo es ahora más conveniente que nunca. Porque agencias como la Agencia Nacional

de Regulación, Vigilancia y Vigilancia Sanitaria (ARCSA) han instado a los profesionales de la salud como lo resalta en la Figura 1, y las farmacias a emitir y aceptar recetas electrónicas, siempre que se cumplan ciertas condiciones y se establezcan ciertas garantías de autenticidad.

Figura 1 Telemedicina (Hep.gob.ec, 2021)

La Organización Mundial de la Salud ha reconocido reiteradamente la contribución de la Telemedicina a la salud y la gestión más eficaz de los sistemas de salud en diferentes países. Sin embargo, este sistema de apoyo nunca ha sido más importante que en la actualidad, porque en el marco de la pandemia, la tecnología hace posible que los profesionales y otros pacientes clasifiquen de forma segura a los pacientes sospechosos de tener Covid-19 y otras enfermedades crónicas.

Además, proporciona una buena alternativa para que los médicos del sector privado atiendan y controlen a sus pacientes, y ninguno de ellos estará expuesto al SARS-CoV-2.

2.2.3 Futuro de la Telemedicina en Ecuador

El Ministerio de Salud Pública, el Ministerio de Telecomunicaciones y la Sociedad de la Información han actualizado recientemente el programa de Telemedicina a nivel nacional, para lo cual esperan implantar este modelo de manera paulatina y objetiva en todas las instituciones públicas de salud.

Por su parte, miembros de la sociedad civil, como el despacho de abogados profesionales de la ley de salud DS Legal Group, han tomado una serie de acciones para buscar establecer un marco legal adecuado para el desarrollo de la Telemedicina en el país. Esto requiere el cumplimiento de regulaciones, permisos de trabajo y operación, y también puede determinar la responsabilidad civil.

2.3. Uso de la Telemedicina mediante el internet de las cosas

Esta tecnología se ha convertido en una realidad en el campo de la salud, de hecho, se ha acuñado el término "Internet médico de las cosas", lo que significa que se puede utilizar en cierta medida para monitorizar enfermedades mediante la captura de señales biológicas. Actualmente, la tecnología RFID se ha utilizado en ensayos clínicos para monitorear a los participantes y es muy útil para la investigación clínica y la epidemiología al proporcionar personal y recursos con datos rastreables en tiempo real, identificación, comunicación, temperatura y ubicación. Todo esto es en esta era en la que la computación ubicua se ha trasladado al campo de la medicina, es decir, la telemetría, en la que las actividades diarias de las personas generan información.

Su uso permitirá el acceso en tiempo real a datos masivos sobre la salud de la población, muchos de los cuales están en tiempo real. Aunque su uso individual puede traer enormes beneficios a la medicina clínica, representa un monitoreo de la salud global a gran escala y otros campos. Prevención, prevención de enfermedades y promoción de la salud, sistemas de seguimiento, sistemas de emergencia, telemedicina e investigación Biomédica (Rodríguez-Gómez, 2019).

2.3.1 Tecnología al servicio de la salud, la epidemiología y la salud pública

La inteligencia artificial ha logrado un progreso increíble en la salud, pero la Telemedicina es una de las áreas más influyentes y la Telemedicina está dando sus frutos en el contexto de la salud pública. Las innovaciones en este campo brindan soluciones reales a los problemas de la Telemedicina, de hecho, está cambiando el desarrollo de nuevos modelos de servicios médicos a través del uso de la evaluación remota, el diagnóstico remoto, la interacción y la monitorización a distancia. Dado que las poblaciones de bajos ingresos y los lugares remotos pueden acceder a determinadas acciones de salud, así como al desarrollo de drones para situaciones específicas, esto provocará una cierta cantidad de daño al sector salud, lo que producirá enormes beneficios, especialmente en ciudades en desarrollo. países. Y las zonas rurales pueden afectar el acceso de las personas a los servicios médicos.

La innovación de la Telemedicina puede acortar la distancia al aumentar el acceso a los servicios médicos para las comunidades desatendidas, aunque esto requiere una fuerte cooperación interdepartamental y una gestión adecuada de los recursos técnicos para lograr una innovación sostenible en la Telemedicina rural. Gracias a la inteligencia artificial, las máquinas son capaces de capturar información del entorno, resolver problemas y adaptarse al entorno, es decir, son inteligentes.

Hoy en día, muchas cosas se clasifican en: teléfonos inteligentes, universidades inteligentes, hogares inteligentes, escuelas inteligentes, estadios inteligentes, fábricas inteligentes, tutores inteligentes, señalización inteligente, iluminación inteligente, autos inteligentes, semáforos y ciudades inteligentes.

Estos sistemas constituyen un nuevo paradigma, frente al actual Internet de las Cosas como representante, millones de dispositivos estarán conectados a Internet y enviarán información diversa sobre innumerables eventos, como una gran cantidad de parámetros biométricos. Personas, condiciones atmosféricas, alertas de terremotos, niveles de contaminación del agua, exposición a radiación ultravioleta, niveles de partículas, calidad del

aire en las casas, monitoreo de la biodiversidad, caducidad de alimentos, accidentes de tránsito, etc. Todo esto está relacionado con la salud pública, ante esta situación se puede hablar de una salud pública inteligente, un entorno digital de vanguardia, en el que la disciplina sea proactiva en la innovación en salud y principalmente en la comprensión, prevención y anticipación de los eventos de salud.

2.4. IoT

La tecnología de Internet de las cosas (IoT) se ha desarrollado continuamente, proporcionando servicios integrales para diversas aplicaciones a través de sus funciones de identificación, recopilación de datos, procesamiento de datos y comunicación rápida. Con el rápido desarrollo de la Internet de las cosas, la tecnología médica moderna se ha optimizado y mejorado, y la aplicación de la Telemedicina se ha vuelto cada vez más extensa. Para monitorear con precisión y recopilar datos de salud del paciente en tiempo real, comparar y analizar la información adquirida y la información médica histórica, y luego brindar a los pacientes servicios médicos oportunos, esta investigación aplicó el Internet de las Cosas y algoritmos inteligentes para recopilar información de telemedicina (Ding, Li, Pan, & Guo, 2021).

El objetivo del Internet de las Cosas es brindar un valor agregado que permita la integración de diferentes tecnologías para generar soluciones que integren diferentes funcionalidades y logren un mejor control de las cosas. Una de las principales razones del desarrollo de la Internet de las Cosas es que los usuarios continúan buscando la interacción con la tecnología, su visión es ir más allá de la escena móvil y convertirse gradualmente en la conexión y combinación de inteligencia y cosas (Gallego López, 2021)

Como se hace referencia en la Figura 2 de la automatización de un hogar.

Figura 2 IoT (Cegasecurity.com, 2021)

El campo de la nueva tecnología ya no será una computadora de escritorio. El Internet de las Cosas permite ubicar cada objeto en la red. En otras palabras, la idea general es que el objeto se pueda conectar a la red para comunicarse con otros dispositivos y ser capaz de cumplir con diferentes predicciones para el entorno de automatización.

Por tanto, a través de diversos protocolos y aplicaciones, en un futuro próximo, la cantidad de dispositivos conectados al ciberespacio o la nube superará las invitaciones de las personas. Por ejemplo, que todo el mundo está rodeado por al menos cien objetos. En el campo de Internet de las cosas, algunas empresas pioneras a nivel mundial incluyen Google, Nidia, Panasonic, Huawei, IBM, etc. Calculan que para 2020, al menos entre 26.000 y 30.000 millones de objetos en el mundo estarán conectados a Internet y, por supuesto, se integrarán las aplicaciones que proporcionará Internet de las cosas.

La Internet de las Cosas puede estimular el desarrollo económico y social y ampliar

los servicios que se pueden proporcionar mediante la popularización y el desarrollo de la tecnología. El uso de las nuevas tecnologías promueve el proceso de comunicación en la sociedad, por lo que se enfatiza la importancia de la investigación, análisis y mejora de las posibles oportunidades de desarrollo, que generan la posibilidad de establecer nuevas ideas de negocio.

2.5. ESP32

ESP32, creado por Espressif Systems, es un sistema de bajo costo y bajo consumo que utiliza una serie de chips (SoC) con capacidades de Wi-Fi y Bluetooth de modo dual. La serie ESP32 incluye los chips ESP32-D0WDQ6 (y ESP32-D0WD), ESP32-D2WD, ESP32-S0WD y ESP32-PICO-D4 system-on-chip (SiP). Esencialmente, hay un microprocesador Tensilica Xtensa LX6 de doble núcleo o de un solo núcleo con una frecuencia principal de 240 MHz. El ESP32 está altamente integrado con el interruptor de antena incorporado, valor de RF, amplificador de potencia, amplificador de receptor de graves, ruido, filtro y potencia. módulo de gestión como se muestra en la Figura 3.

El ESP32 está diseñado específicamente para dispositivos móviles, dispositivos electrónicos portátiles y aplicaciones de IoT. Logra un consumo de energía ultra bajo a través de características de ahorro de energía. Estas características incluyen sincronización de reloj de alta resolución, múltiples modos de consumo de energía y escalado dinámico de energía.

Una MCU rica en funciones con conectividad Wi-Fi y Bluetooth incorporada, adecuada para varias aplicaciones (espressif.com, 2021).

Figura 3 (Amazon.com, 2021)

2.5.1. Características del ESP32

El ESP32 puede funcionar de forma fiable en entornos industriales a temperaturas de funcionamiento que oscilan entre –40 ° C y + 125 ° C. El ESP32 funciona con un circuito de calibración avanzado, que puede eliminar dinámicamente los defectos del circuito externo y adaptarse a los cambios en las condiciones externas.

ESP32 está especialmente diseñado para dispositivos móviles, dispositivos electrónicos portátiles y aplicaciones de IoT. Combina varios tipos de software propietario para lograr un consumo de energía ultra bajo. ESP32 también incluye las funciones más avanzadas, como sincronización de reloj detallada, varios modos de potencia y escalado dinámico de potencia.

ESP32 está altamente integrado con interruptor de antena incorporado, balun de RF, amplificador de potencia, amplificador receptor de bajo ruido, filtro y módulo de administración de energía. ESP32 agrega funcionalidad y versatilidad valiosas a su aplicación con requisitos mínimos de placa de circuito impreso (PCB).

ESP32 se puede utilizar como un sistema independiente completo o como un dispositivo esclavo del MCU anfitrión, reduciendo así la sobrecarga de la pila de comunicación en el procesador de la aplicación principal. ESP32 puede conectarse con otros sistemas a

través de su interfaz SPI / SDIO o I2C / UART para proporcionar funciones de Wi-Fi y Bluetooth.

2.6. AWS IOT

AWS IoT proporciona servicios en la nube que conectan su dispositivo IoT a otros dispositivos y servicios en la nube de AWS como se muestra en la Figura 4. AWS IoT proporciona software para dispositivos que puede ayudarlo a integrar dispositivos IoT con soluciones basadas en AWS IoT. Si sus dispositivos pueden conectarse a AWS IoT, AWS IoT puede conectarlos a los servicios en la nube proporcionados por AWS. Servicios de IoT para soluciones empresariales, industriales y de consumo. (Aws.amazon.com, 2021)

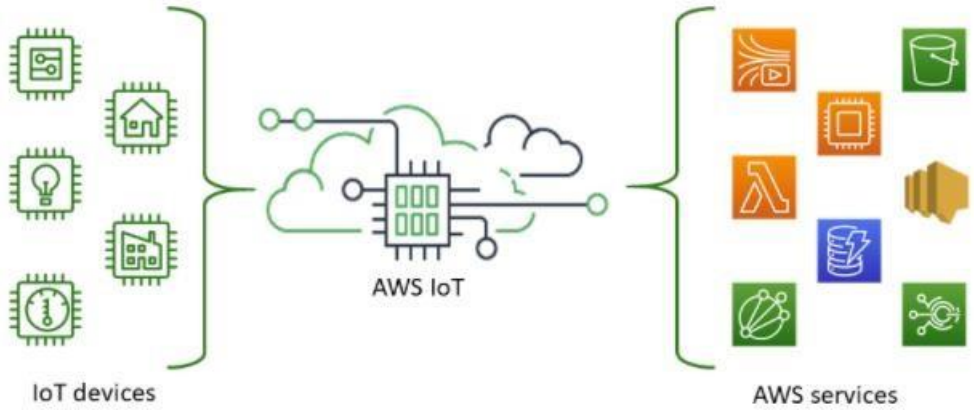


Figura 4 AWS IOT (Aws.amazon.com, 2021)

AWS IoT le permite elegir la tecnología más adecuada y reciente para su solución. Para ayudarlo a administrar y admitir dispositivos de IoT en el campo, AWS IoT Core admite los siguientes protocolos:

- MQTT (Message Queue Server y transporte de telemetría)

- MQTT sobre WSS (Websockets Secure)

- HTTPS (Protocolo de transferencia de hipertexto - Seguro).

- LoraWan (Red de área amplia de largo alcance)

2.6.1. Características de AWS

Desde ubicaciones de borde hasta la nube, AWS tiene una amplia gama de poderosos servicios de IoT. AWS IoT es el único proveedor en la nube que combina análisis y administración de datos enriquecidos en servicios fáciles de usar diseñados para datos ruidosos de IoT.

WS IoT proporciona servicios para todas las capas de seguridad, incluidos los mecanismos de seguridad preventiva (como el control de acceso y el cifrado de datos del dispositivo) y un servicio para la supervisión y auditoría continuas de las configuraciones.

AWS está combinando IA e IoT para hacer que los dispositivos sean más inteligentes. Puede crear modelos en la nube e implementarlos en dispositivos, y se ejecutan dos veces más rápido que otros servicios del mercado.

AWS IoT se basa en una infraestructura de nube segura y probada que puede escalar a miles de millones de dispositivos y billones de mensajes. AWS IoT está integrado con otros servicios de AWS, por lo que puede crear una solución completa.

2.7. AWS DynamodB

Amazon DynamoDB es una base de datos multirregional duradera y totalmente administrada con respaldo, recuperación y seguridad integrados, así como una memoria caché para aplicaciones a escala de Internet. DynamoDB puede manejar más de 10 billones

de solicitudes por día y puede soportar un pico de más de 20 millones de solicitudes por segundo. Cientos de miles de clientes de AWS eligen DynamoDB como su valor clave y base de datos de documentos para aplicaciones móviles, web, de juegos, tecnología publicitaria y de IoT, así como más aplicaciones que necesitan acceder a datos de baja latencia de cualquier escala. (DynamoDB, 2020).

DynamoDB proporciona funciones de copia de seguridad bajo demanda. Permite crear copias de seguridad completas de tablas para archivarlas y retenerlas a largo plazo para cumplir con los requisitos de cumplimiento normativo.

2.7.1. Beneficios de DynamoDB

DynamoDB admite algunas de las aplicaciones más grandes del mundo y proporciona tiempos de respuesta de milisegundos de un solo dígito de cualquier tamaño. Puede crear aplicaciones con capacidad de procesamiento y almacenamiento casi ilimitada. DynamoDB Global Tables replica sus datos en varias regiones de AWS, lo que le permite acceder rápida y localmente a los datos desde aplicaciones distribuidas globalmente. Para casos de uso que requieren un acceso más rápido con latencia de microsegundos, DynamoDB Accelerator (DAX) proporciona una caché en memoria completamente administrada.

DynamoDB expandirá o encogerá automáticamente la mesa para ajustar la capacidad y mantener el rendimiento. Funciones integradas de usabilidad y tolerancia a fallos, por lo que no necesita considerar estas funciones al diseñar su aplicación. DynamoDB proporciona capacidad bajo demanda y modelos de suministro de capacidad para que pueda optimizar los costos especificando la capacidad por carga de trabajo o pagando por los recursos que consume.

2.7.2. Aplicaciones

- **Aplicaciones web sin servidor**

Crea aplicaciones web eficientes que se puedan escalar automáticamente. No necesita mantener servidores y aplicaciones con alta disponibilidad automatizada.

- **Back-ends móviles**

Utiliza DynamoDB y AWS AppSync para crear aplicaciones web y móviles interactivas con actualizaciones en tiempo real, acceso a datos sin conexión y sincronización de datos con resolución de conflictos incorporada.

- **Microservicios**

Crea microservicios flexibles y reutilizables utilizando DynamoDB como un almacén de datos sin servidor para obtener un rendimiento ágil y estable.

2.8. AWS TimeStream

Amazon Timestream es un servicio de base de datos de series de tiempo rápido, escalable y sin servidor para aplicaciones operativas y de IoT. Le permite almacenar y analizar fácilmente billones de eventos por día a una velocidad 1000 veces mayor que el costo. Amazon Timestream le ahorra el tiempo y el costo de administrar el ciclo de vida de los datos de series de tiempo al almacenar los datos más recientes en la memoria y mover los datos históricos a una capa de almacenamiento con costos optimizados según las estrategias definidas por el nombre del usuario

El motor de consultas personalizadas de Amazon Timestream como lo indica en la figura 5 le permite acceder y analizar datos combinados recientes e históricos sin tener que especificar explícitamente en la consulta si los datos residen en la memoria o en una capa de costo optimizado. (Amazon Timestream, 2020)

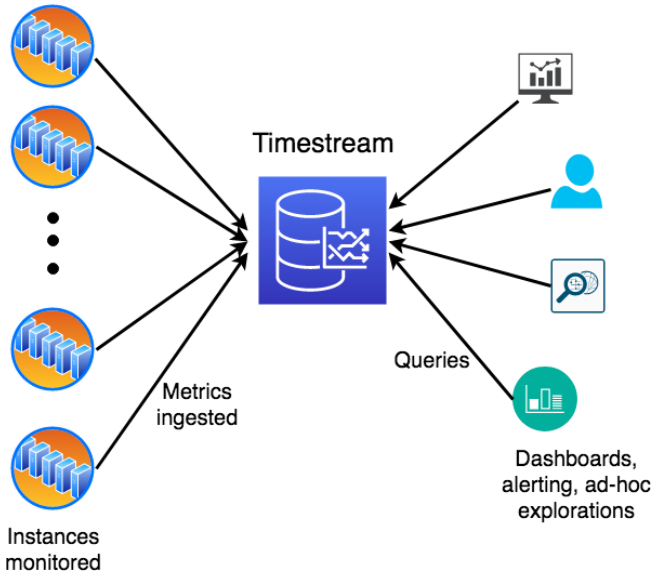


Figura 5 Amazon Timestream (AWS Database Blog, 2020)

2.8.1. Beneficios de Amazon Timestream

- Permite la incorporación de datos de alto capacidad, consultas rápidas en un momento determinado a través de su almacenamiento en memoria y consultas analíticas rápidas a través del almacenamiento magnético de costo optimizado.

- Amazon Timestream le ofrece la escala para procesar billones de eventos y millones de consultas por día. A medida que su aplicación necesite

cambiar, se escala automáticamente para ajustar la capacidad.

- El motor de consultas especialmente diseñado de Amazon Timestream acceder de forma transparente y combina datos en distintas capas de almacenamiento sin que usted tenga que especificar la ubicación de los datos.

2.9. UBidots

Plataforma encargada del ensamble y ejecución rápidamente aplicaciones de Internet de las cosas (IoT) sin tener que escribir código o contratar un equipo de desarrollo de software. Las decisiones se pueden tomar con las herramientas o API que se tienen para la visualización y el análisis de datos. Maneja diferentes protocolos de conexión, entre los que se pueden mencionar principalmente HTTP, MQTT y TCP. (Ubidots, 2020)

2.9.1. Beneficios de Ubidots

- Desarrollo de aplicaciones en la nube de apuntar y hacer clic.

- Desarrolle las soluciones de su empresa con las herramientas de desarrollo de aplicaciones intuitivas y sin código de Ubidots: motor de reglas y eventos, informes programados, paneles de control en tiempo real y más.

- Aplicaciones específicas del cliente de marca blanca.

- Implemente aplicaciones personalizadas para los usuarios finales con la marca, las URL y los colores de su empresa, además de personalizar los mensajes de

alerta, los idiomas y más para ofrecer sus soluciones de la manera que desee.

- Conecte fácilmente hardware y fuentes de datos a la nube.

- Conecte su hardware y / o servicios de datos digitales a la nube independiente de dispositivos de Ubidots con nuestra API REST y brinde soluciones personalizadas de IoT y nube de la manera que desee.

2.10. Integración AWS y Ubidots

Ubidots y AWS se han asociado para crear una integración prediseñada que permite a los usuarios reenviar fácilmente los datos de los sensores desde AWS IoT Core a Ubidots, utilizando complementos como se visualiza en la Figura 6. (Ubidots plugins, 2020)

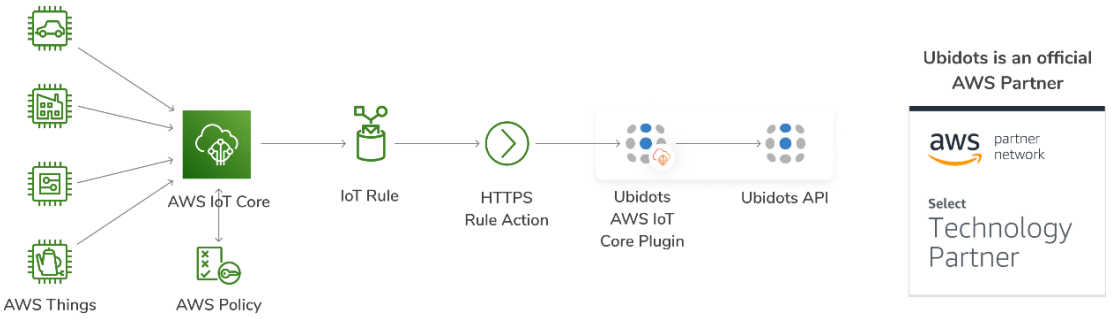


Figura 6 integración AWS Iot con Ubidots (Ubidots plugins, 2020)

Esta integración creará automáticamente los siguientes recursos en su cuenta de AWS:

- Una acción de regla de AWS HTTPS IoT
- Un destino de regla HTTPS confirmado

2.11. A20 Olinuxino Micro

Es una computadora de placa única de hardware abierto capaz de ejecutar Android o Linux diseñada por OLIMEX Ltd. en Bulgaria. El objetivo del proyecto era diseñar una placa Linux de grado industrial amigable con el bricolaje que todos puedan reproducir en casa. Aprovecha los componentes soldables a mano ampliamente disponibles que son razonables para comprar en pequeñas cantidades, alojados en paquetes TQFP. Los archivos CAD del proyecto están alojados en GitHub, lo que permite a todos estudiarlos y personalizarlos según sus necesidades. Inicialmente, Olinuxino fue diseñado con EAGLE. En marzo de 2016, las primeras placas diseñadas con KiCad estuvieron disponibles cuando OLIMEX Ltd. anunció planes para cambiar el desarrollo a herramientas CAD de código abierto (Olimex, 2021a).

Características:

- Soc.: A20 Dual Core Cortex-A7 CPU 1GHz
- GPU: Mali 400 GPU
- RAM: 1024MB DDR3
- SATA con salida para alimentar a 5V el disco SATA
- HDMI conector con soporte FullHD (1080p)
- 2 x USB High-speed host con control de potencia y limitador de corriente
- USB-OTG con control de potencia y limitador de corriente
- Gigabit 1000MBit native Ethernet
- Conexión para batería LiPo con control de carga
- LED de estado de la batería
- LCD conector compatible con pantallas de 4.3", 7.0', 10.1' de su marca Olimex

- 160 GPIOs en cuatro GPIO conectores
- Lector de tarjetas MicroSD para el Sistema Operativo
- Posibilidad de memoria interna de 4GB para el Sistema Operativo.

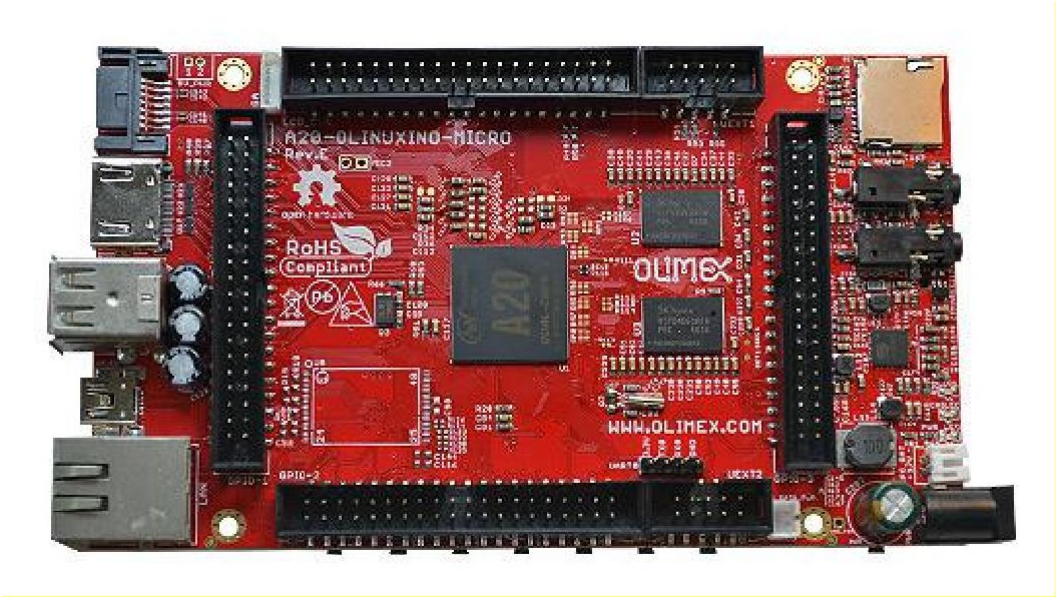


Figura 7 A20-Olinuxino-MICRO (Olimex, 2021)

2.12. Shield EKG olinuxino

SHIELD-EKG-EMG convierte la señal diferencial analógica (los potenciales biológicos ECG / EMG generados por los músculos), unida a sus entradas CH1_IN + / CH1_IN-, en una sola secuencia de datos como salida.

La señal de salida es analógica y debe discretizarse aún más con el objetivo de ofrecer la opción de procesamiento digital. Por lo general, esto se realiza a través de ADC dedicado integrado en la MCU de la placa base (como: OLIMEXINO-328, OLIMEXINO-32U4, OLIMEXINO-STM32) (Olimex, 2021c).

A continuación, descripción gráfica de Olinuxino EKG con pines de conexión:

Figura 8 Shield-Ekg-Emg (Olimex, 2021)

2.13. MOD-EKG Olimex

MOD-EKG es una implementación del diseño de hardware propuesto por Texas Instruments en su documento "Heart-Rate and EKG Monitor Using the MSP430FG439" (nombre del documento "slaa280a"). MOD-EKG es un proyecto de hardware y software abiertos. El usuario coloca sus dedos índices y pulgares en las almohadillas metálicas de la placa y la frecuencia del latido del corazón por minuto se muestra en la pantalla LCD. Además, la aplicación envía un flujo de datos digital a una interfaz de datos serie que le permite mostrar la forma de onda EKG en un PC. La interfaz serie está disponible en el conector UEXT.

Usando nuestro adaptador USB de ↔ serie. Una conexión entre el PC y MOD-EKG se puede lograr mediante el uso de nuestro adaptador – MOD-USB-RS232 que convierte la interfaz serie a USB, y viceversa.

Un electrocardiograma (ECG), también llamado ECG, es un gráfico que muestra el voltaje generado por el músculo cardíaco durante un latido del corazón. En esta aplicación, la forma de onda EKG es utilizada por el MCU para medir la frecuencia del latido del corazón. Debido a que el cálculo de los latidos del corazón es el enfoque principal, los electrodos se simplifican a dos puntos para recopilar datos, uno para la mano derecha y el otro para la mano izquierda (Olimex, 2021b)

A continuación, descripción gráfica del MOD-EKG Olimexcon pines de conexión:

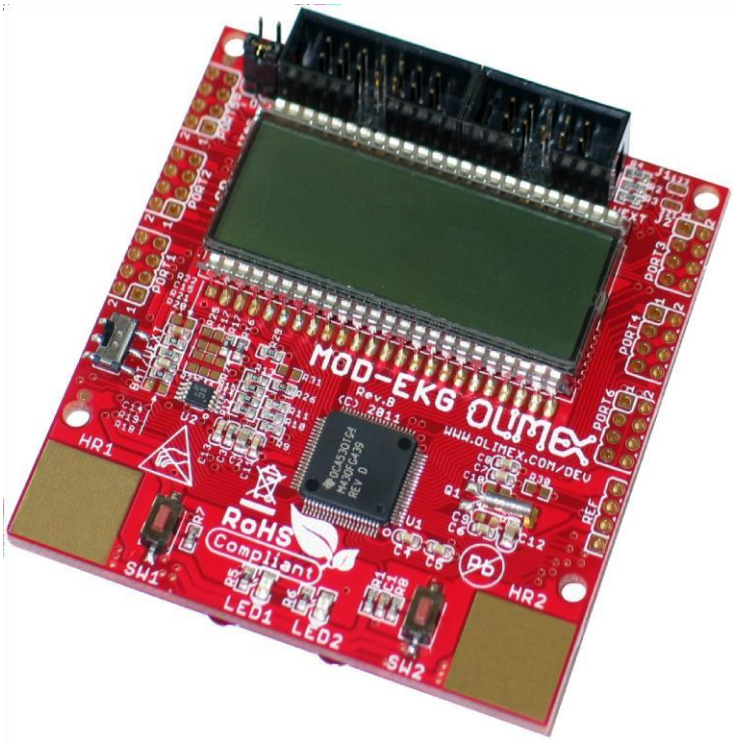


Figura 9 MOD Ekg Olimex(Olimex, 2021)

2.13.1. Características del MOD EKG Olimex

- Microcontrolador: MSP430FG439
- Amplificadores EKG con INA321EA
- Conector JTAG paso a paso de 14 pines y 0,1" para el acceso directo del microcontrolador a través del depurador MSP430 para depurar el software o alimentar la placa
- Conector UEXT que permite una conexión más fácil a otras placas
- Pantalla LCD personalizada
- Soporte de batería de moneda de litio CR2032 de 3V
- Filas de extensión de agujeros de alfiler para todos los puertos de microcontrolador, sin conectores de plástico colocados
- ¡No se requieren electrodos adicionales! Electrodos montados en tablero de contacto de la mano derecha e izquierda – HR1 y HR2
- Dos LED de estado • Dos botones de usuario
- Un interruptor para alternar entre la fuente de alimentación "batería" y "UEXT"
- Basado en la nota de aplicación de Texas Instruments SLAA280a
- FR-4, 1,5 mm, soldermask rojo, impresión de componentes blancos
- Dimensiones: (3.0 x 2.8)" ~ (76 x 71) mm
-

2.13.2. Diseño de la placa MOD-EKG Olimex

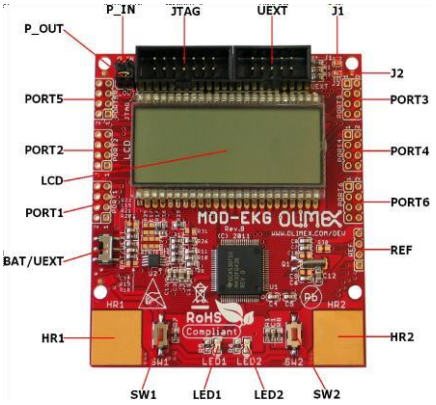


Figura 10 Pines del MOD Ekg Olimex(Olimex, 2021)

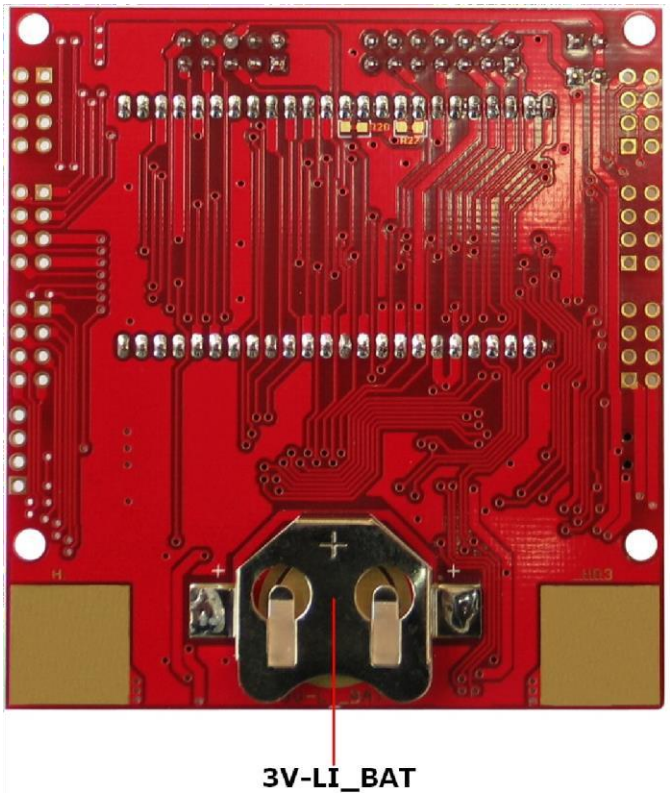


Figura 11 Conexión de batería MOD Ekg Olimex(Olimex, 2021)

2.14. Cámara ESP32

Esta placa de desarrollo es una placa basada en ESP32 con WiFi y Bluetooth y tiene

una cámara incorporada. Además, la placa también tiene una conexión para tarjeta microSD.

Figura 12 Cámara ESP32

Especificaciones:

- Chip ESP32 con procesador de doble núcleo de 240 MHz
- Memoria flash: 4 MB
- Memoria de trabajo: 512 KB SRAM, 4 MB PSRAM
- Cámara: OV2640
- Hasta 10 pines digitales
- Hasta 7 pines analógicos

2.15. Sensor de temperatura corporal

Este sensor como se visualiza en la Figura 13 viene con una placa de conexión con todos los componentes necesarios para el funcionamiento y dos tipos de pines. Utiliza la luz infrarroja para medir la temperatura, incluso de objetos remotos sin la necesidad de estar en contacto con ellos. Fuente de alimentación: 3-5v (regulador interno de baja tensión).

Comunicación: Protocolo de comunicación estándar de la IIC.

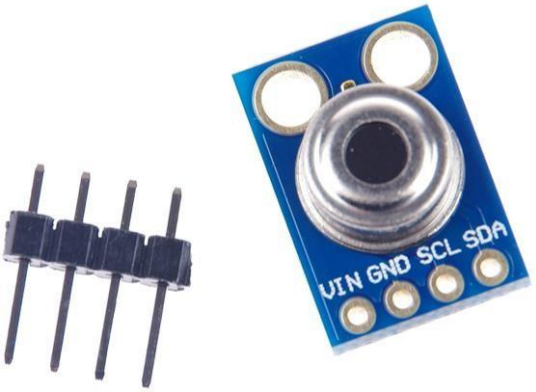


Figura 13 Sensor de temperatura corporal

2.15.1. Características y beneficios:

- Calibrado de fábrica en amplio rango de temperatura: -40 a 125 °C para temperatura del sensor y -70 a 380 °C para temperatura del objeto.
- Alta precisión de 0,5 °C sobre un amplio rango de temperatura (0..+50 C tanto para Ta como To). Precisión médica de 0,1 °C en un rango de temperatura limitado disponible bajo petición.
- Resolución de medición de 0,02 °C.
- Versiones de una y dos zonas.
- Interfaz digital compatible con SMBus para lecturas rápidas de temperatura y redes de sensores de construcción.
- Salida PWM personalizable para una lectura continua.
- Disponible en versiones de 3 V y 5 V.
- Adaptación sencilla para aplicaciones de 8 a 16 V.
- Modo de ahorro de energía.
- Diferentes opciones de paquete para aplicaciones y medidas versátiles.
- Grado automotriz.

2.16. Oxímetro

Es un dispositivo que integra un pulsioxímetro y un pulsómetro como se visualiza en la Figura 14. Tiene dos LED: un LED rojo (660nm) y un LED infrarrojo (920nm), un fotodetector, ópticas especiales, un filtro de luz ambiental entre 50 y 60 Hz, y un conversor ADC delta sigma de 16 bits, hasta 1000 muestras por segundo. Además, tiene un sensor de temperatura interno para compensar el efecto de la temperatura en la medición. La longitud de onda de la luz emitida es lo suficientemente precisa, es decir, sigue las longitudes de ondaroja (650 nm) e infrarroja (950 nm), y el sensor que detecta la luz es lo suficientemente precisocomo para determinar la saturación (porcentaje) de saturación de oxígeno en sangre arterial.

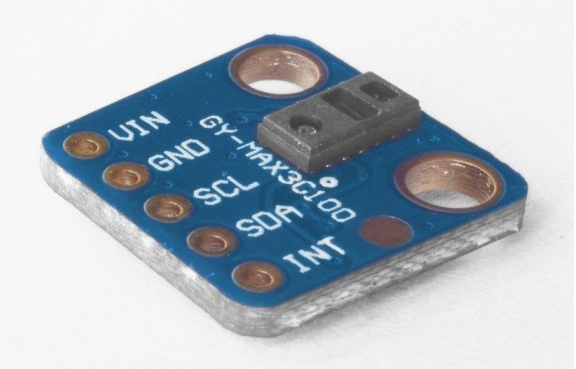


Figura 14 Oxímetro

2.16.1. Especificaciones técnicas

- Voltaje de Operación: 5V DC
- Regulador de voltaje de 3.3V y 1.8V en placa
- Led rojo de 660nm
- Led infrarrojo de 920nm
- Filtro de luz entre 50 y 60Hz
- Protocolo de comunicación: I2C
- ADC delta sigma de hasta 16 bits
- Temperatura de trabajo: -40°C hasta +85°C

- Dimensiones: 14mm x 17mm

2.17. Balanza digital Redmi Xiaomi

Permite el seguimiento de los cambios en su cuerpo y el registro de los datos con la aplicación. Viene con la función de prueba de peso ideal para ajustar el peso adecuado de acuerdo con el IMC y su edad. Envía los datos de peso y control a la aplicación. Además, también permite calcular el Índice de Masa Corporal (IMC). Permite un peso máximo de hasta 150 Kg.

Especificaciones:

- Material protector: Cristal Templado
- Material: ABS
- Peso: 1.2Kg
- Color: blanco
- Dimensiones: 280mm x 280mm x 22mm
- Rango: De 0.05Kg a 150Kg

La báscula inteligente Xiaomi está hecha de plástico ABS y el panel principal está diseñado con vidrio templado fuerte y suave. Para evitar posibles resbalones y caídas, sobre todo cuando se lo presiona con los pies mojados, su superficie tiene un revestimiento antideslizante, allí también se encuentran cuatro electrodos de acero inoxidable.

A diferencia de otros modelos en el mercado, esta báscula no muestra una pantalla, pero la oculta hasta tres segundos después de subir al automóvil. En este momento, la lectura del peso se mostrará a través de la luz LED (o la batería está baja y ha superado los 150 kg. Entonces el peso máximo) como se visualiza en la Figura 15. En este sentido, incluye un

sensor de luz que puede ajustar automáticamente el brillo de estas luces para leer mejor los números. (Xiaomi, 2021)

Figura 15 Báscula Xiaomi (Xiaomi, 2021)

3. Marco metodológico

3.1. Tipo de investigación

El proyecto se basa en una investigación exploratoria, que busca una visión general en el campo de la Telemedicina.

Se realiza una investigación experimental porque las variables de estudio se obtienen a partir de las prácticas realizadas con los manuales de Telemedicina propuesto en este trabajo de investigación, los experimentos y pruebas se realizarán en los laboratorios de la Universidad Politécnica Salesiana Sede Guayaquil; por este motivo, es necesario realizar un análisis de cada uno de los laboratorios propuestos para el banco de pruebas de

Telemedicina. (Arturo et al., 2011)

3.2. Diseño de investigación

3.3. Metodología de investigación

La metodología por utilizar es la experimental, para este proyecto se realizará la adquisición de datos mediante los sensores del EKG, báscula digital, temperatura corporal, oxímetro y cámara web para monitoreo remoto, los mismos que a través de un microcontrolador como el ESP32 y WIFI enviaran datos a través de la plataforma de AWS IoT para ser procesados y gestionados de manera tal que se pueda desarrollar prácticas de Telemedicina así como también fomenta la investigación y prototipados a nivel de Telemedicina.

Trabajos relacionados se citan a continuación:

Diseño Y Construcción De Un Electrocardiógrafo de 12 Derivaciones Para El Análisis De Señales Cardiacas”

Autor: Guillermo Eduardo Vega Picón.

Universidad Politécnica Salesiana.

“Desarrollo De Un Sistema De Adquisición Y Tratamiento de Señales Electrocardiográficas”

Autor: Cristian Vidal Silva, Leopoldo Pavesi Farriol.

Universidad Tarapac.

“Electrocardiógrafo inteligente de bajo costo”

Autor: Alexander Mariel Mendiguren.
Universidad del país vasco.

3.4. Descripción de la propuesta del prototipo IoT de Telemedicina

Se propone un prototipo IoT de telemedicina para módulo didáctico para prácticas con estudiantes de la Carrera de Electrónica y Automatización y la Carrera de Ingeniería de Telecomunicaciones.

Como se puede observar en la figura 16, la propuesta se basa en bancos de pruebas basado en microcontrolador ESP32 y el AD8232 cual procesará todos los datos obtenidos mediante el sensor EKG y serán enviados a través de una red inalámbrica al internet para que

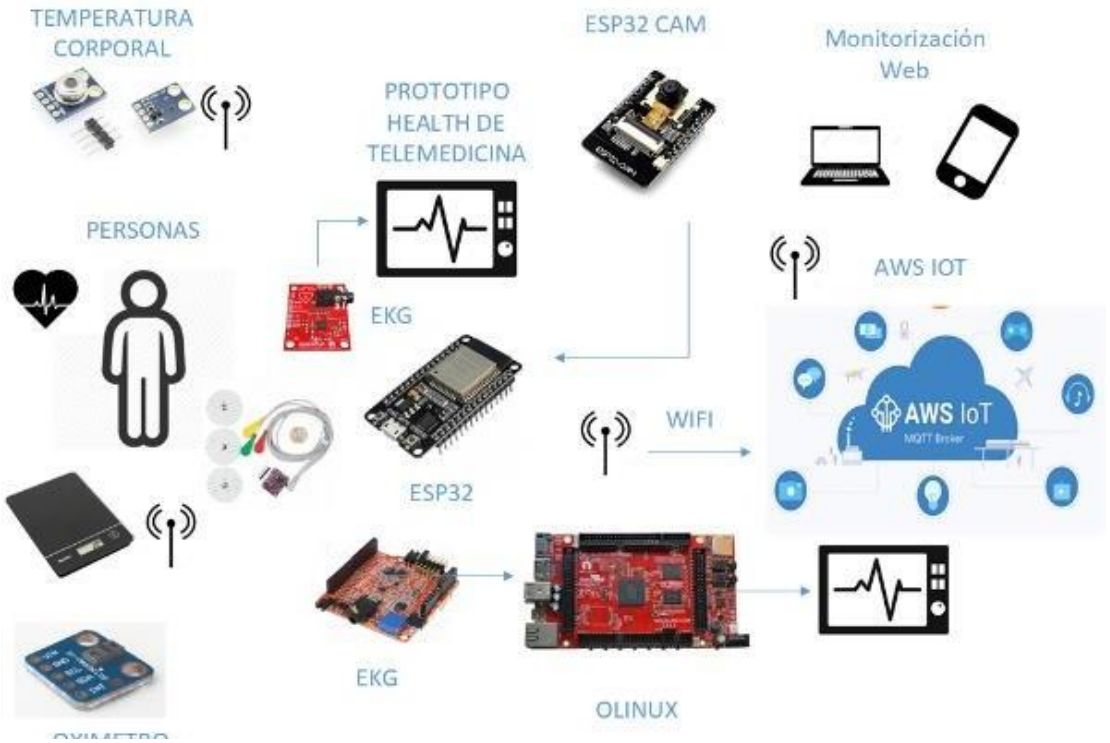


Figura 16 Microcontrolador ESP32 y el AD8232

estos datos sean almacenados y tratados en servidores web en la nube como lo es AWS Amazon Web Services y Ubidots, aplicaciones de IoT (internet de las cosas). De la misma manera se contará con otro módulo basado en Olimex A20 y sensores EKG de Olimex que mostrarán los datos del EKG de manera local, contando así con módulos prácticos de Telemedicina para el desarrollo de investigación y prácticas en los estudiantes de las Carreras de Ingeniería en Electrónica y Telecomunicaciones.

Se propone un banco de pruebas de seis prácticas las cuales se detalla a continuación:

Práctica #1: Diseño e implementación de banco de prueba de Telemedicina con electrocardiograma de 3 vías con SHIELD-EKG-EMG y A20 Olinuxino Micro.

Práctica #2: Diseño e implementación de banco de prueba de Telemedicina con electrocardiograma de 3 vías EKG, ESP32 y AD8232.

Práctica #3: Diseño e implementación de banco de prueba de Telemedicina con báscula digital inalámbrica con conexión al ESP32.

Práctica #4: Diseño e implementación de banco de prueba de Telemedicina con oxímetro digital con conexión al ESP32.

Práctica #5: Diseño e implementación de banco de prueba de Telemedicina con sensor de temperatura corporal y conectividad al ESP32.

Práctica #6: Configuración de servicios web en la nube con Amazon Web Service de prototipo de banco de pruebas de Telemedicina e implementación de cámara web con ESP32 para monitoreo visual remoto.

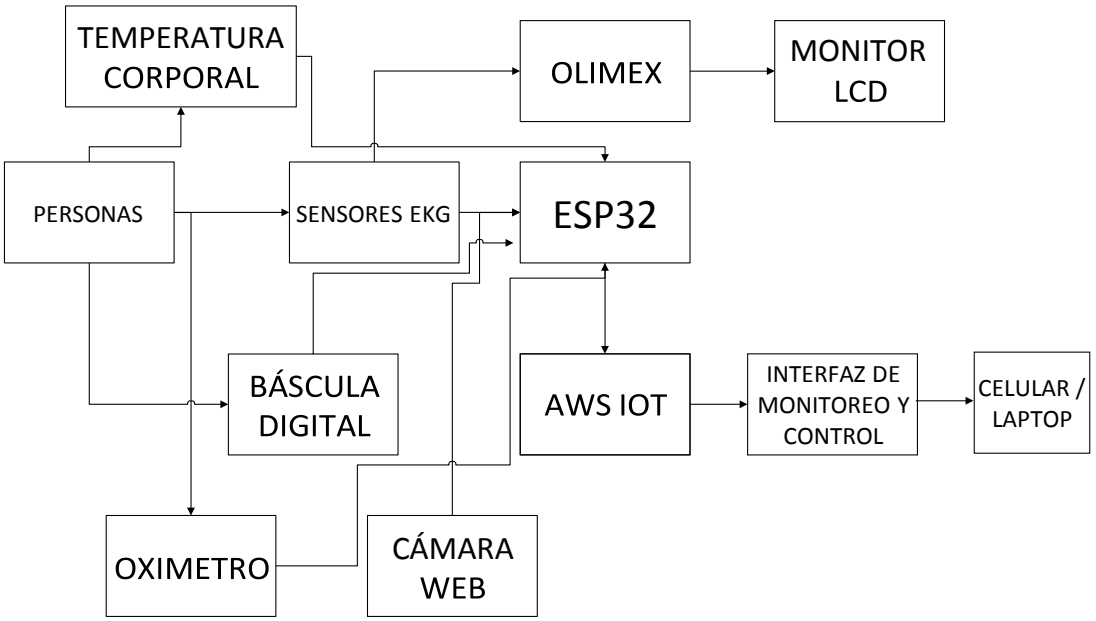


Figura 17 Diagrama de bloques de prototipo IoT de Telemedicina

4. Resultados

En este capítulo se explicará las configuraciones mas importantes realizadas para el funcionamiento del prototipo de Telemedicina IoT y los resultados obtenidos.

Finalmente se detallan los objetivos de cada práctica propuesta en este trabajo.

4.1. Diagrama de conexiones de la tarjeta PCB

En la figura 18. se muestra el datasheet de los controles de los sensores que están implementados en el prototipo de telemedicina.

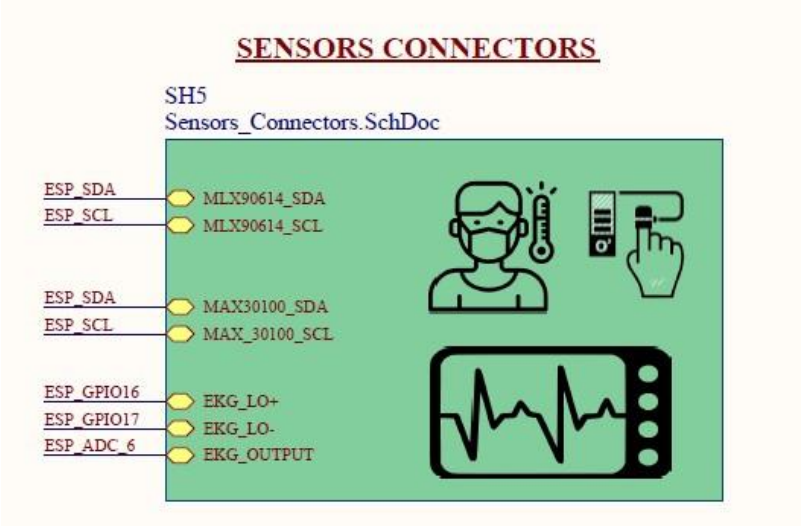


Figura 18 Conector de sensores

Se muestra en la figura 19. los diferentes tipos de reguladores y la protección que existe entre 5+ y +3.3 VDC que se utilizan en el prototipo y la protección de la energía.

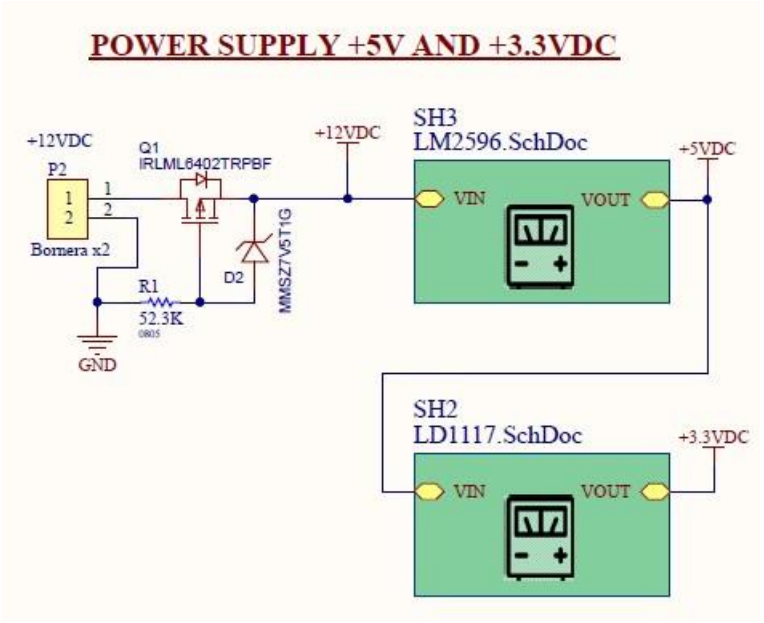


Figura 19 Reguladores y protección

En la figura 20. se muestra el datasheet del componente USB To UART Bridge con el cual se trabaja para el prototipo de Telemedicina, para convertir USB a Serial.

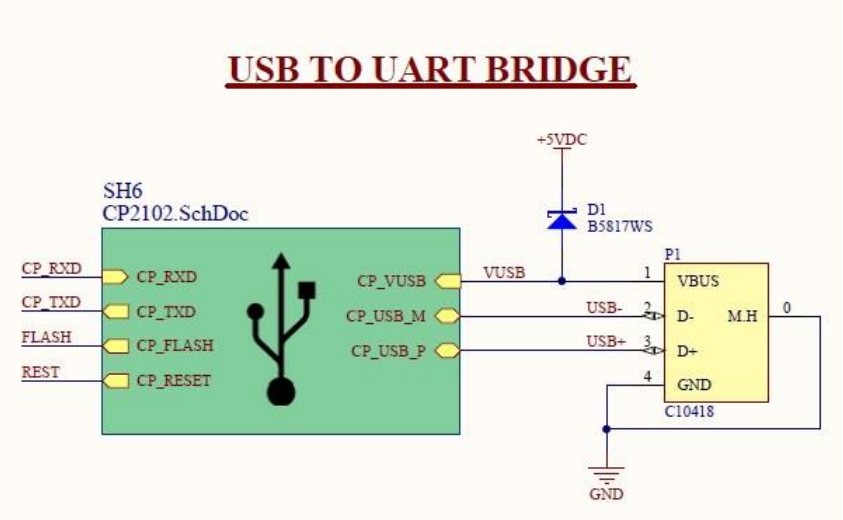


Figura 20 Convertidor USB a Serial

En la figura 21. se muestra el datasheet del componente ESP32 Wi-fi Core con el que se implementa para el prototipo aplicado a Telemedicina.

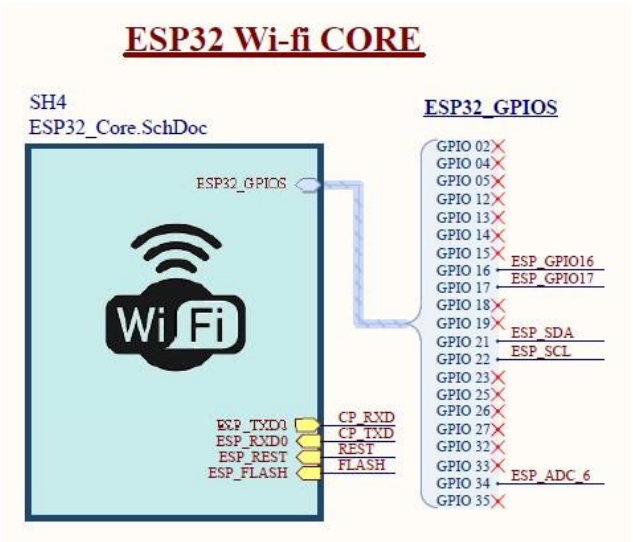


Figura 21 ESP32 Wifi Core

En la figura 22. se observa el esquemático del USB to UART Bridge, convertidor de USB a Serial.

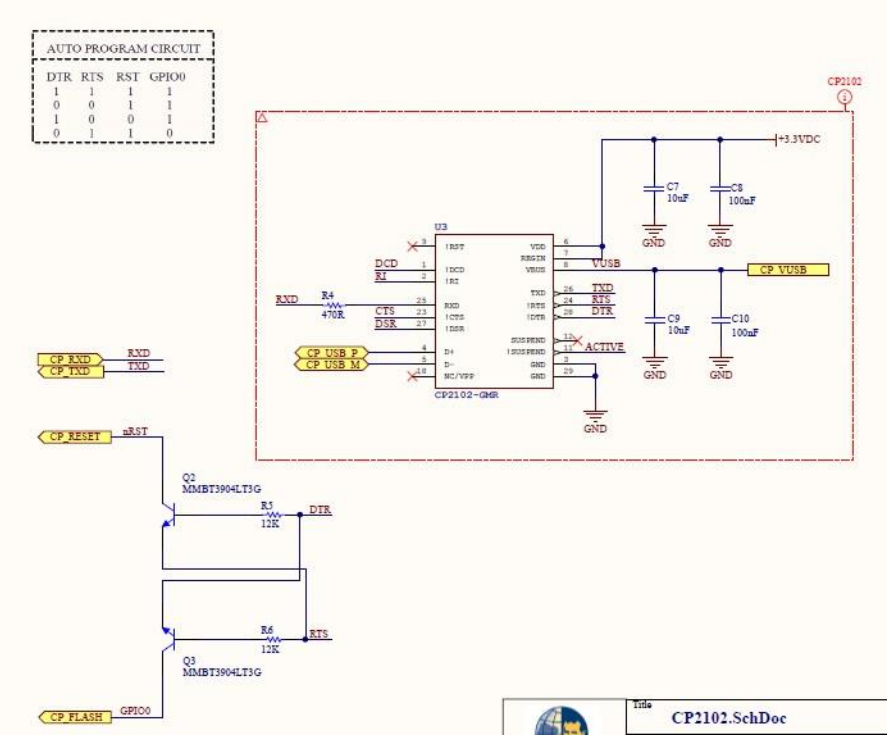


Figura 22 Esquemático del CP2102-GMR

En la figura 23. se muestra el esquemático del microchip LD1117ADT33TR, el cual regula la de tensión LDO 3.3V 1.0A.

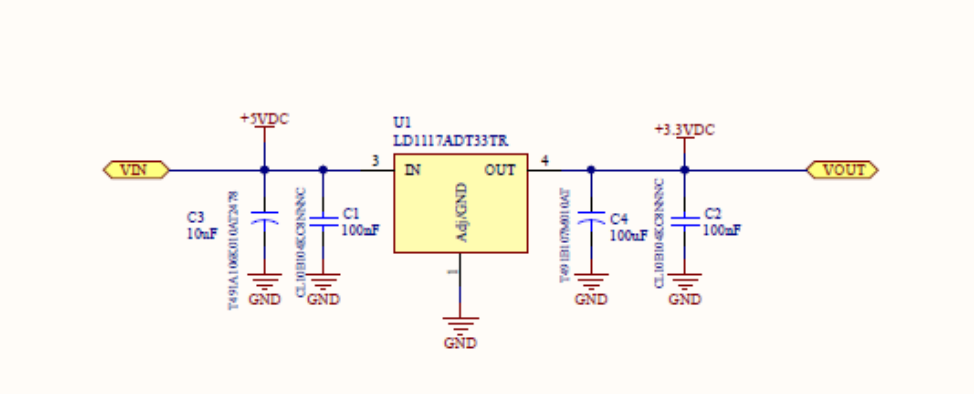


Figura 23 Esquemático de LD1117ADT33TR

En la siguiente figura 24. se muestra el esquemático del LM2576, siendo el regulador de voltaje de conmutación 3A.

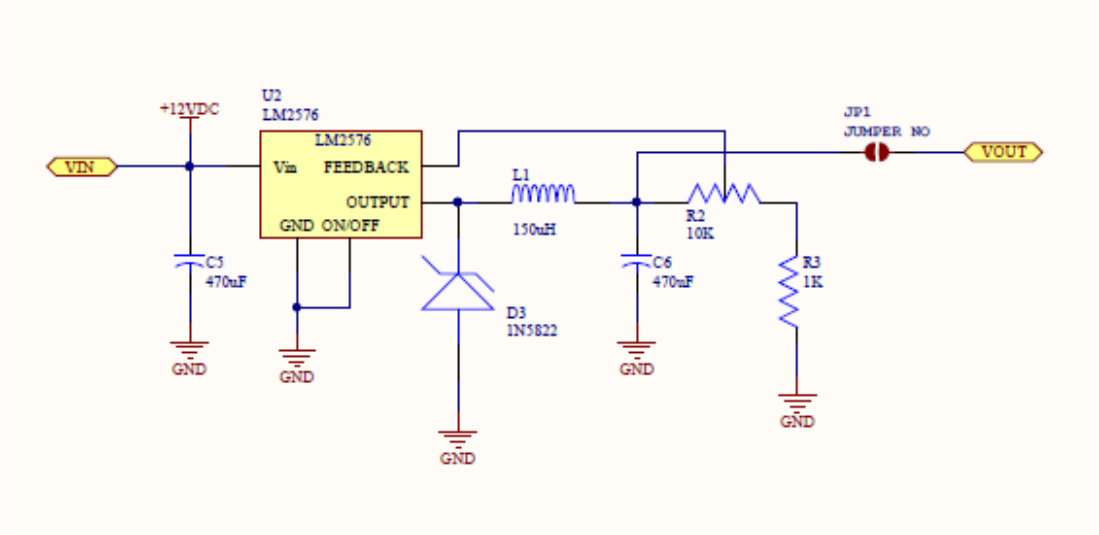


Figura 24 Esquemático del LM2576

En la siguiente figura 25. se muestra el esquemático del ESP32, Módulo Wifi (802.11).

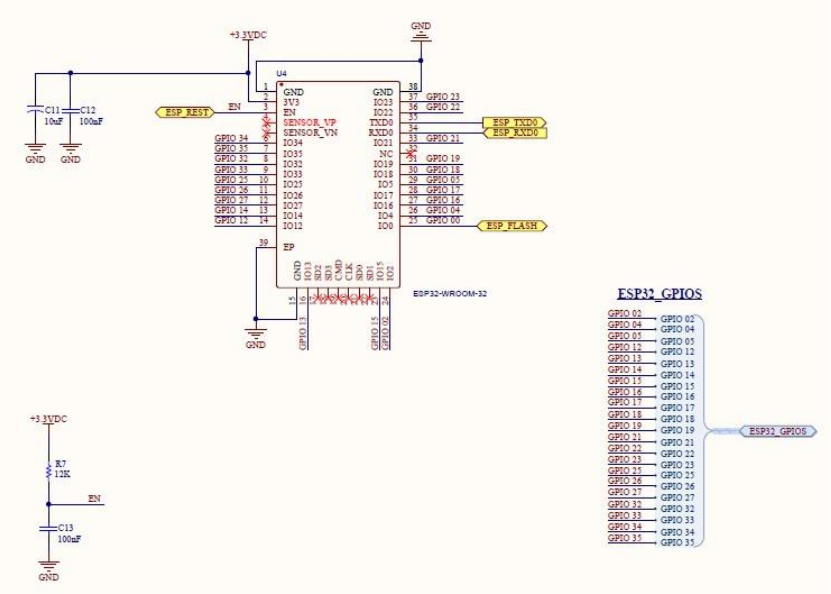


Figura 25 Esquemático del ESP32

En la figura 26. se muestra el esquemático de las conexiones que existen en los sensores del prototipo de telemedicina.

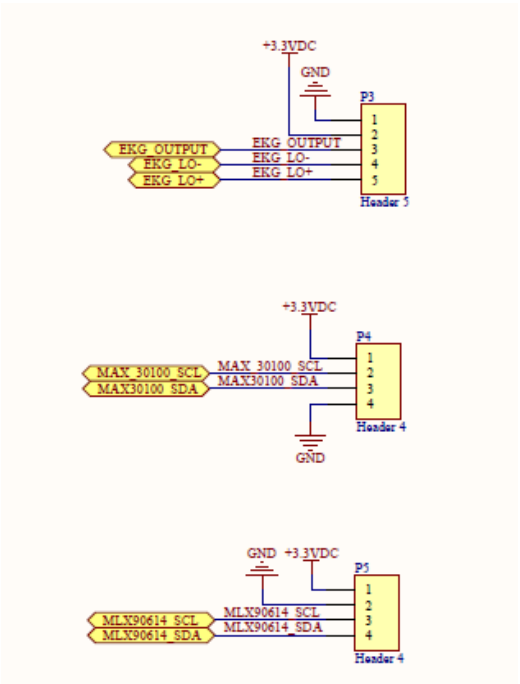


Figura 26 Esquemático de conexión

En la figura 27. se observa el diseño de la placa electrónica con los diferentes componentes implementados.

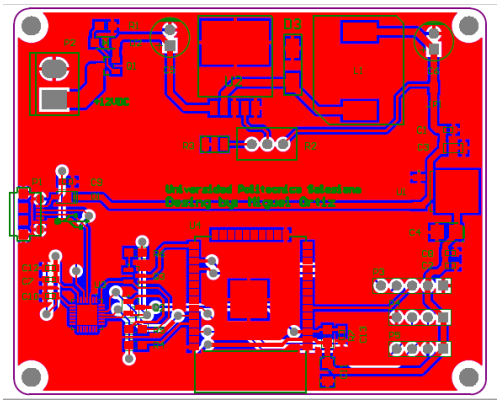


Figura 27 Diseño de la placa electrónica

En la figura 28. se muestra la parte frontal del diseño de la placa electrónica en 3D del prototipo de Telemedicina.

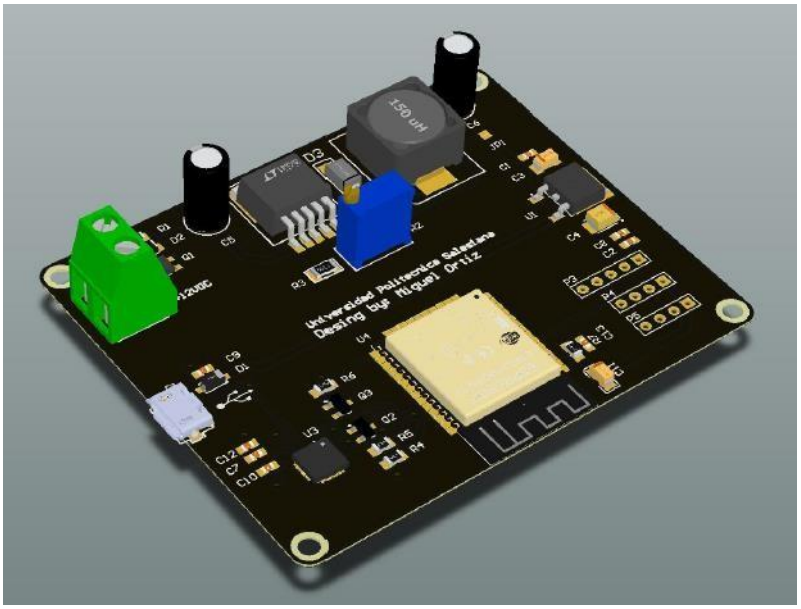


Figura 28 Frontal de placa electrónica

4.2. Diseño de la carcasa en 3D

En las figuras 29, 30, 31,32 y 33 se muestran las imágenes de las carcasas impresas en 3D los cuales protegerán los diseños electrónicos y las placas del módulo y de los elementos del prototipo de Telemedicina IoT.

Figura 29 Carcaza principal del prototipo de Telemedicina IoT.

Figura 30 Vista lateral de conectores del prototipo de Telemedicina IoT.

Figura 31 Vista frontal de las cajas protectoras del prototipo Iot Telemedicina.

Figura 32 Carcaza de protección de la tarjeta A20 Olinuxino Micro

Figura 33 Vista lateral de conexión de periféricos de A20 Olinuxino Micro.

4.3. Ensamblaje de PCB en carcaza 3D

En la figura 34. se observa las placas que se utilizan en la implementación de los sensores para el prototipo.

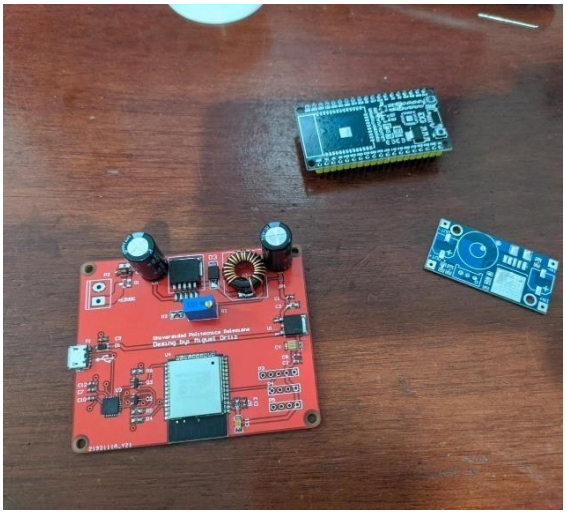


Figura 34 Placas electrónicas de los diferentes sensores.

4.4. Elementos y sensores del prototipo de Telemedicina

A continuación se muestran las conexiones físicas de los elementos y sensores del prototipo de Telemedicina IoT.

La figura 35. muestra la conexión de los electrodos ECG al case principal del prototipo de Telemedicina.

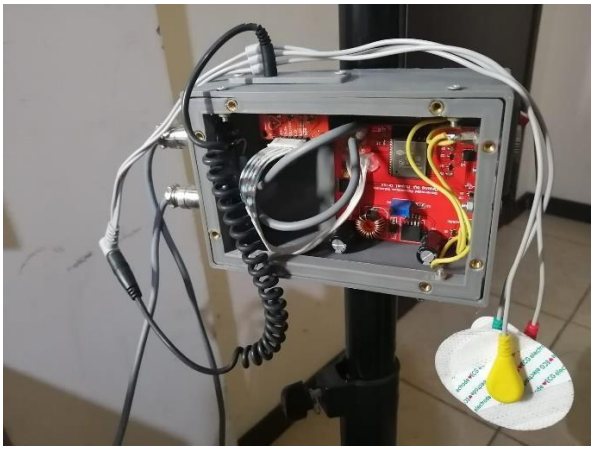


Figura 35 Conexión de electrodos en ECG

La figura 36. muestra el case en 3D del sensor de temperatura infrarojos.

Figura 36 Sensor de temperatura con infrarrojo

La figura 37 muestra el sensor de pulso con su respectivo case protector realizado en impresora 3D.

Figura 37 Sensor de pulso y oxímetro.

La figura 38. muestra el módulo de conexión USB RS232 de Olimex que se utiliza para la comunicación serie con la tarjeta MOD EKG de Olimex y un PC.

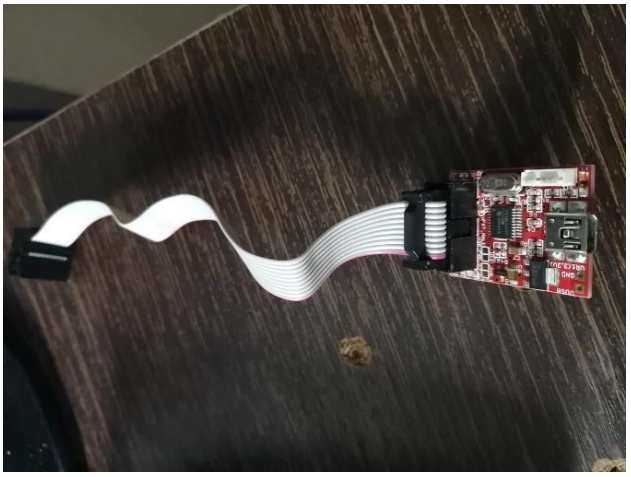


Figura 38 Módulo de conexión USB RS232

La figura 39. muestra los electrodos de 3 vías que se conectará al prototipo IoT.

Figura 39 Sensores electrodos ECG

Figura 40 Cámara IP ESP32Cam

Figura 41 Báscula inalámbrica

4.5. Práctica #1: Diseño e implementación de banco de prueba de Telemedicina con electrocardiograma de 3 vías con MOD-EKG y A20 Olinux.

Objetivo general:

Aprender el uso del banco de pruebas de Telemedicina con el electrocardiograma de 3 vías Shield EKG EMG y A20 Olinuxino Micro

Objetivos específicos:

• Comprender el funcionamiento de placa A20 Olinuxino Micro.
• Realizar la programación para visualización de EKG con Shield EKG EMG
• Realizar pruebas con MOD-EKG Olimex.

4.6. Práctica #2: Diseño e implementación de banco de prueba de Telemedicina con electrocardiograma de 3 vías EKG, ESP32 y AD8232.

Objetivo general:

Aprender el uso del banco de pruebas de Telemedicina con el electrocardiograma de 3 vías EKG, ESP32 y AD8232

Objetivos específicos:

- Realizar configuración de EKG con ESP32.
- Visualizar datos desde IDE de Arduino.
- Visualizar datos del EKG en Ubidots.

4.7. Práctica #3: Diseño e implementación de banco de prueba de Telemedicina con báscula digital inalámbrica con conexión al ESP32.

Objetivo general:

Aprender el uso del banco de pruebas de Telemedicina con báscula digital inalámbrica con conexión al ESP32.

Objetivos específicos:

- Conectorizar báscula digital al ESP32.
- Monitorizar datos de báscula digital.
- Enviar información de báscula digital a Ubidots.

4.8. Práctica #4: Diseño e implementación de banco de prueba de Telemedicina con oxímetro digital con conexión al ESP32.

Objetivo general:

Aprender el uso del banco de pruebas de Telemedicina con oxímetro digital con conexión al esp32.

Objetivos específicos:

• realizar la configuración de oxímetro digital
• visualizar la información del oxímetro en servidores web de Amazon.
• visualizar la información del oxímetro en servidores web de Ubidots.

4.9. Práctica #5: Diseño e implementación de banco de prueba de Telemedicina con sensor de temperatura corporal y conectividad al ESP32.

Objetivo general:

Aprender el uso del banco de pruebas de telemedicina con sensor de temperatura corporal y conectividad al ESP32.

Objetivos específicos:

• Realizar configuración de sensor de temperatura infrarrojos.
• Visualizar los datos de temperatura en servicios de AWS.
• Visualizar datos de temperatura en servicios de Ubidots.

4.10. Práctica #6: Configuración de servicios web en la nube con Amazon Web Service de prototipo de banco de pruebas de Telemedicina e implementación de cámara web con ESP32 para monitoreo visual remoto.

Objetivo general:

Aprender la configuración de servicios web en la nube con Amazon Web Service de prototipo de banco de pruebas de Telemedicina e implementación de cámara web con ESP32 para monitoreo visual remoto.

Objetivos específicos:

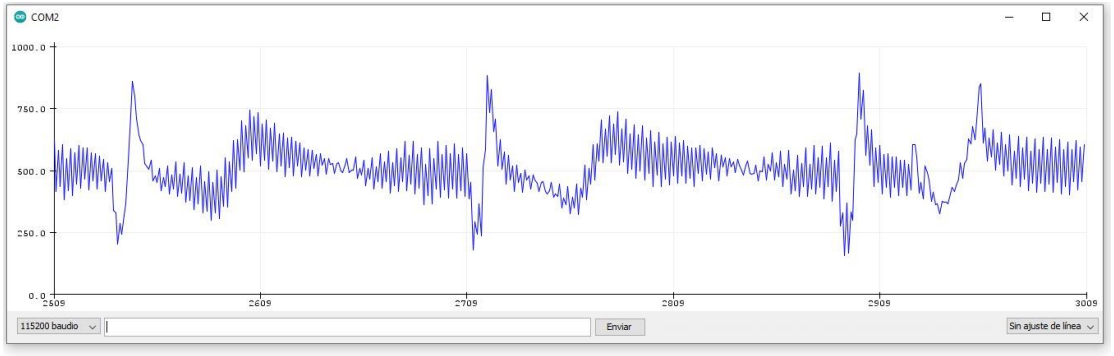
• Conectorizar cámara ESP32 para visualización de pacientes que usarán el prototipo de Telemedicina.

• Subir datos adquiridos con los sensores principales a AWS

• Almacenar en DynamoDB los datos obtenidos con el prototipo de telemedicina.

5. Análisis de resultados

Se detalla a continuación un resumen de los resultados obtenidos al ejecutar cada una de las prácticas. Cabe mencionar que el procedimiento en detalle de cada práctica se encuentra en la sección de Anexos.



5.1. Práctica #1: Diseño e implementación de banco de prueba de Telemedicina con electrocardiograma de 3 vías con MOD-EKG y A20 Olinux.

En esta práctica se realiza el diseño e implementación de un banco de pruebas de Telemedicina con electrocardiograma de 3 vías con modulo Shield EKG de Olimex y tarjeta A20 Olinuxino. Al integrar estas tarjetas al banco de pruebas se valida el correcto funcionamiento de un EKG educativo de Telemedicina con el cual se obtienen datos del ritmo cardiaco del paciente y los datos son subidos a servidores web en la nube con el cual se validan los objetivos planteados en la práctica. El detalle de esta práctica se encuentra en los anexos del libro.

5.2. Práctica #2: Diseño e implementación de banco de prueba de Telemedicina con electrocardiograma de 3 vías EKG, ESP32 y AD8232.

En esta práctica se realiza el diseño e implementación de un banco de pruebas de Telemedicina con electrocardiograma de 3 vías con EKG, ESP32 y AD8232. La tarjeta principal es el AD8232 el cual obtiene los datos del EKG y lo envía a Ubidots para la visualización de la gráfica del ritmo cardiaco del paciente con el cual se validan los objetivos planteados en la práctica. El detalle de esta práctica se encuentra en los anexos del libro.

5.3. Práctica #3: Diseño e implementación de banco de prueba de Telemedicina con báscula digital inalámbrica con conexión al ESP32.

En esta práctica se realiza el diseño e implementación de un banco de pruebas de Telemedicina con báscula digital inalámbrica y conexión al ESP32. Se utiliza la báscula MiLite 2 de Xiaomi el cual se conecta al ESP32 y se obtiene los datos del peso del paciente, esta información luego es enviada a Ubidots para ser monitorizada de forma remota. El detalle de esta práctica se encuentra en los anexos del libro.

5.4. Práctica #4: Diseño e implementación de banco de prueba de Telemedicina con oxímetro digital con conexión al ESP32.

En esta práctica se realiza el diseño e implementación de un banco de pruebas de Telemedicina con oxímetro digital y conexión al ESP32, en esta práctica los resultados se obtienen al conectorizar los sensores al paciente y obtener los datos de oxigenación, y pulsómetro, los datos son enviados al ESP32 el cual enviará la data Ubidots para monitorizar la información en los servicios web de la nube. El detalle de esta práctica se encuentra en los anexos del libro.

5.5. Práctica #5: Diseño e implementación de banco de prueba de Telemedicina con sensor de temperatura corporal y conectividad al ESP32.

En esta práctica se realiza el diseño e implementación de un banco de pruebas de Telemedicina con sensor de temperatura corporal y conexión al ESP32, en esta práctica los resultados se obtienen al utilizar el sensor de temperatura corporal que está fabricado en forma de pistola con luz infrarroja el cual obtendrá los datos de temperatura del paciente. Esta información es enviada al ESP32 el cual posteriormente enviará la data a los servicios en nube de Ubidots para su monitorización y almacenamiento. El detalle de esta práctica se encuentra en los anexos del libro.

5.6. Práctica #6: Configuración de servicios web en la nube con Amazon Web Service de prototipo de banco de pruebas de Telemedicina e implementación de cámara web con ESP32 para monitoreo visual remoto.

En esta práctica se realiza la configuración de servicios web en la nube con Amazon Web Service del prototipo de banco de pruebas de Telemedicina e implementación de cámara web con ESP32 para monitoreo visual remoto. Se tiene como resultado en esta práctica los respaldos en Amazon Bucket S3 de todas las mediciones realizadas en las prácticas anteriores, además se incluye la Cámara ESP32Cam para la visualización remota del paciente.

El detalle de esta práctica se encuentra en los anexos del libro.

6. Conclusiones

Las conclusiones de este trabajo de investigación son las siguientes:

- Se realiza el diseño e implementación de un módulo didáctico para aplicaciones de Telemedicina con servicios web en la nube e internet de las cosas (IoT), para esto se utilizan elementos como el ESP32 y sensores de bajo costo para armar el prototipo, como servicios web se utiliza Ubidots para la monitorización y visualización de los datos, y como respaldo de los datos se utilizan servicios de Amazon Bucket S3 de Amazon Web Services.

- Se realiza el diseño de un módulo didáctico de electrocardiograma EKG con Olimex A20 y Shield EKG y electrodos de 3 vías. Para el cumplimento de este objetivo se utilizan elementos de Olimex y Shield ESPduino para realiza la toma de datos y

enviarlos a servicios en la nube de Ubidots.

- Se realiza el diseño de un módulo didáctico de electrocardiograma EKG con ESP32 y AD8232 y electrodos de 3 vías. Para el cumplimiento de este objetivo se utiliza el módulo AD8232 el cual es el encargado de tomar los datos de los sensores del EKG y enviará la data al ESP32 para su posterior envió a los servidores de Ubidots.

- Se realiza la configuración de los servicios en la nube para el almacenamiento y monitorización de la información de los EKG con Amazon Web Services, para el cumplimiento de este objetivo se configuraron los servicios en Ubidots para la integración con AWS y poder realizar los respaldos con Amazon S3.

- Se realiza el manual de seis prácticas didácticas del prototipo IoT de Telemedicina el cual se adjuntan en el anexo de este libro. En cada manual de práctica se detallan los procedimientos para la correcta configuración y pruebas de cada práctica.

7. Recomendaciones

Las recomendaciones de este trabajo de investigación son las siguientes:

- Se debe considerar que este prototipo de Telemedicina es de uso exclusivamente académico, no se puede utilizar para determinar alguna enfermedad o conclusión médica, ya que los elementos utilizados en el prototipo lo especifican en su descripción técnica.

- Los datos obtenidos con el EKG de Olimex y el EKG del AD8232 son muestras gráficas aproximadas de un EKG real y certificado, no se debe considerar utilizar estas muestras para un análisis médico de cardiología. Es importante acotar que las gráficas subidas a la web para monitorizar por Ubidots pueden tener valores no precisos ya que la latencia en una red de internet influirá en la gráfica vista en

Ubidots. Se recomienda realizar la monitorización desde el panel Serial Plotter del IDE de Arduino para unas gráficas de mejor calidad.

- Los datos obtenidos con la balanza Xiaomi son datos reales que se suben a la nube mediante Ubidots, por lo tanto, este dato si es válido para un registro posterior de cada paciente.

- Los datos obtenidos por el sensor de temperatura corporal y el oxímetro son datos precisos que se suben a Ubidots para un posterior análisis.

- La cámara web ESP32Cam debe estar conectado a una red inalámbrica de buena calidad con pocas interferencias para poder enviar el video streaming al internet y verlo desde una ip publica con su respectivo redireccionamiento de puertos.

- Para una correcta monitorización de los datos en Ubidots se recomienda adquirir el plan de Udidots IoT Entrepreneur, el cual otorga más capacidad de datos hasta para 25 dispositivos y 2 millones de datos por mes.

- Para realizar las pruebas de cada práctica se recomienda cargar el código compartido en cada práctica y probar los códigos por separado, estos códigos son modificables si el docente o el estudiante requiera obtener otros resultados o agrupar las prácticas.

Bibliografía

Amazon.com. (2021). Amazon.com: HiLetgo ESP-WROOM-32 ESP32 ESP-32S junta de desarrollo 2.4 GHz WiFi + Bluetooth modo dual dos núcleos procesador de microcontrolador integrado con antena amplificadora de RF Filtro AP STA Arduino IDE: Computers & Accessories. Retrieved May 18, 2021, from https://www.amazon.com/HiLetgo-ESP-WROOM-32-Development-Microcontroller-Integrated/dp/B0718T232Z

Arturo, C., Álvarez, M., Surcolombiana, U., De, F., Sociales, C., Humanas, Y., … Monje Álvarez, C. A. (2011). *METODOLOGÍA DE LA INVESTIGACIÓN CUANTITATIVA Y CUALITATIVA Guía didáctica*.

Aws.amazon.com. (2021a). ¿Qué es AWS IoT? - AWS IoT Core. Retrieved May 18, 2021, from https://docs.aws.amazon.com/es_es/iot/latest/developerguide/what-is-aws-iot.html

Aws.amazon.com. (2021b). ¿Qué es AWS IoT? - AWS IoT Core.

Cegasecurity.com. (2021). Internet de las Cosas IOT - CEGA Security. Retrieved May 18, 2021, from https://cegasecurity.com/casos-de-uso/internet-de-las-cosas-iot/

Consumotic.mx. (2021). IoT, el poder tecnológico al servicio de la salud | Consumotic. Retrieved May 18, 2021, from https://www.consumotic.mx/tecnologia/iot/iot-el-poder-tecnologico-al-servicio-de-la-salud/

Ding, T., Li, J., Pan, J., & Guo, D. (2021). Human remote mobile medical information collection method based on internet of things and intelligent algorithm. *Revista Brasileira de Medicina Do Esporte*, *27*(Special issue), 28–30. https://doi.org/10.1590/1517-8692202127012020_0091

Domingues, R. B., Mantese, C. E., Aquino, E. da S., Fantini, F. G. M. M., Fernandes do Prado, G., & Nitrini, R. (2020). Telemedicine in neurology: Current evidence. *Arquivos de Neuro-Psiquiatria*, *78*(12), 818–826. https://doi.org/10.1590/0004-282X20200131

DynamoDB, A. (2020). AWS | Servicio de base de datos gestionada NoSQL (DynamoDB).

Edicionmedica.ec. (2021). La Telemedicina debe ser regulada formalmente en Ecuador. Retrieved May 18, 2021, from https://www.edicionmedica.ec/secciones/profesionales/la-telemedicina-debe-ser-regula-formalmente-en-ecuador--96207

espressif.com. (2021). Módulos ESP32 Wi-Fi y Bluetooth I Espressif. Retrieved May 18, 2021, from https://www.espressif.com/en/products/modules/esp32

Gallego López, D. H. (2021). *Monografía introductoria en los sistemas IoT con énfasis en los sectores de la salud, la educación y la agro-industria*.

Hep.gob.ec. (2021). Telemedicina | Hospital de Especialidades Portoviejo | Ecuador. Retrieved May 18, 2021, from https://www.hep.gob.ec/telemedicina-3/

khn.org. (2021). Telemedicina: guía para entender las citas médicas a distancia | Kaiser Health News. Retrieved May 18, 2021, from https://khn.org/news/telemedicina-guia-para-entender-las-citas-medicas-a-distancia/

Mesa, M., & Iván Pérez, H. (2020a). The medical act in the era of telemedicine. *Revista Medica de Chile*, *148*(6), 852–857. https://doi.org/10.4067/S0034-98872020000600852

Mesa, M., & Iván Pérez, H. (2020b). The medical act in the era of telemedicine. *Revista Medica de Chile*, *148*(6), 852–857. https://doi.org/10.4067/S0034-98872020000600852

Olimex. (2021a). A20-OLinuXino-MICRO - Open Source Hardware Board. Retrieved August 9, 2021, from https://www.olimex.com/Products/OLinuXino/A20/A20-OLinuXino-MICRO/open-source-hardware

Olimex. (2021b). MOD-EKG - Open Source Hardware Board. Retrieved August 9, 2021, from https://www.olimex.com/Products/Modules/Biofeedback/MOD-EKG/open-source-hardware

Olimex. (2021c). SHIELD-EKG-EMG - Open Source Hardware Board. Retrieved August 9, 2021, from https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/open-source-hardware

Ouali, A., Poon, K. F., Lee, B. S., & Romaithi, K. Al. (2016). Towards achieving practical GPON FTTH designs. *2015 IEEE 20th International Workshop on Computer Aided Modelling and Design of Communication Links and Networks, CAMAD 2015*. https://doi.org/10.1109/CAMAD.2015.7390490

Raj, C., Jain, C., & Arif, W. (2018). HEMAN: Health monitoring and nous: An IoT based e-health care system for remote telemedicine. *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017*, *2018-January*, 2115–2119. https://doi.org/10.1109/WiSPNET.2017.8300134

Rodríguez-Gómez, R. (2019). Internet de las cosas: Futuro y desafío para la epidemiología y la salud pública. *Universidad y Salud*, *21*(3), 253–260. https://doi.org/10.22267/rus.192103.162

Saludiario.com. (2021). Regulación legal de la telemedicina en Ecuador | Saludiario. Retrieved May 18, 2021, from https://www.saludiario.com/regulacion-legal-de-la-telemedicina-en-ecuador/

Seo.com.ec. (2021). La Telemedicina, una pieza clave en el futuro de la salud | Sociedad Ecuatoriana de Oncología - SEO. Retrieved May 18, 2021, from https://seo.com.ec/2020/04/20/la-telemedicina-una-pieza-clave-en-el-futuro-de-la-salud/

Ubidots. (2020). Plataforma IoT | Internet de las cosas | Ubidots.

Ubidots plugins. (2020). Plugins: Connect AWS IoT Core to Ubidots using HTTPS| Centro de ayuda de Ubidots.

Xiaomi. (2021). Xiaomi Mi Body Composition Scale, análisis: review con características y precio. Retrieved August 9, 2021, from https://www.xataka.com/analisis/xiaomi-mi-body-composition-scale-analisis-caracteristicas-precio-especificaciones

Anexos

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR		**GUÍA DE PRÁCTICA DE LABORATORIO**	
CARRERA: INGENIERÍA ELECTRÓNICA			**ASIGNATURA**:
NRO.		**TÍTULO PRÁCTICA**: Diseño e implementación	

PRÁCTICA:		de banco de prueba de Telemedicina con electrocardiograma de 3 vías con SHIELD-EKG-EMG y A20 Olinuxino Micro.

OBJETIVO GENERAL:

Aprender el uso del banco de pruebas de Telemedicina con el electrocardiograma de 3 vías Shield EKG EMG y A20 Olinuxino Micro

OBJETIVOS ESPECÍFICOS:

- Comprender el funcionamiento de placa A20 Olinuxino Micro.
- Realizar la programación para visualización de EKG con Shield EKG EMG
- Realizar pruebas con MOD-EKG Olimex.

INSTRUCCIONES	1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo.
	2. Los estudiantes deben utilizar los equipos del banco de pruebas de telemedicina de una manera responsable y calificada para evitar daños en los equipos.
	3. Los estudiantes deben trabajar en grupo para el desarrollo de la práctica.
	4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta.

ACTIVIDADES POR DESARROLLAR:

1) Se realiza un diseño e implementación de un prototipo de monitoreo de Telemedicina con Olimex.

2) Montar y realizar la programación para las señales obtenidas con el prototipo y poder ser visualizadas con Shield EKG EMG

3) Realizar las pruebas del de funcionamiento para garantizar el funcionamiento óptimo del prototipo.

4) En la figura 1 se muestra la conexión de los periféricos como monitor, teclado y mouse a la tarjeta Olimex A20, ya que esta tarjeta actúa como un PC basado en Linux para la monitorización del EKG con Shield Olimex.

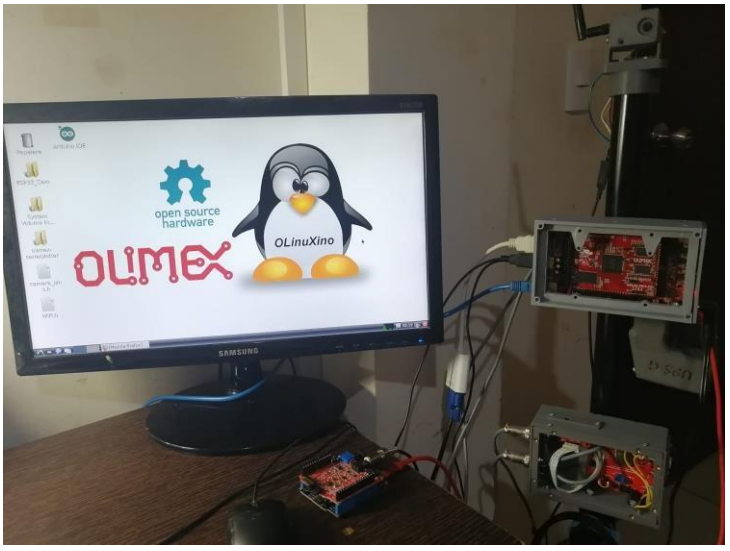


Figura 1: Montaje del prototipo diseñado

5) Para este diseño se utiliza la tarjeta Olimex A20 Olinuxino – Micro.

Figura 2: Carcaza de placa olimex

6) Instalación de tarjeta Olimex A20 en caja impresa en 3D.

Figura 3: Caja con OLIMEX A20

7) Conexión de Shield Olimex con ESPduino.

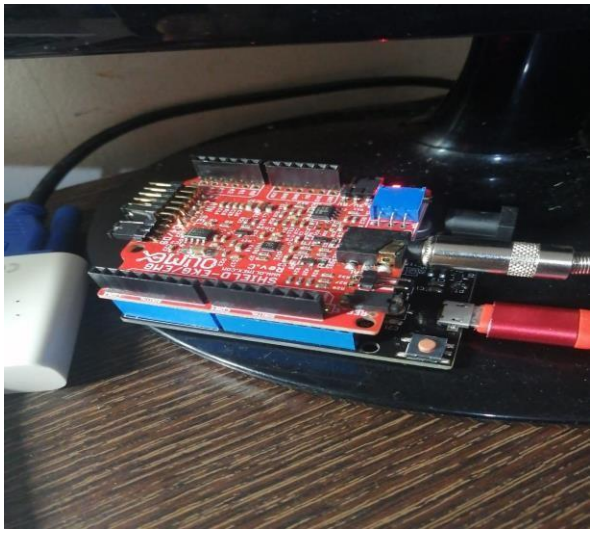


Figura 4: Conexión de Shield Olimex EKG y ESPduino

8) A continuación, se muestra el código que se graba en el ESPduino y Shield Olimex EKG

Código cargado en IDE de Arduino para monitoreo local.

```
//UNIVERSIDAD POLITÉCNICA SALESIANA - SEDE GUAYAQUIL
//INGENIERÍA EN ELECTRÓNICA
//PROTOTIPO DE TELEMEDICINA - IOT
//AUTORES: MIGUEL ORTIZ - CRISTIAN MACIAS

// LAB1 - EKG_OLIMEX

void setup() {
  Serial.begin(115200);
}
```

```
void loop() {
  int sensorValue = analogRead(A0);
  // print out the value
  Serial.println(sensorValue);
  // about 256Hz sample rate
  delayMicroseconds(4100);
}
```

9) Luego de haber cargado el código al ESPDuino se configura la visualización del EKG desde el Olimex A20.

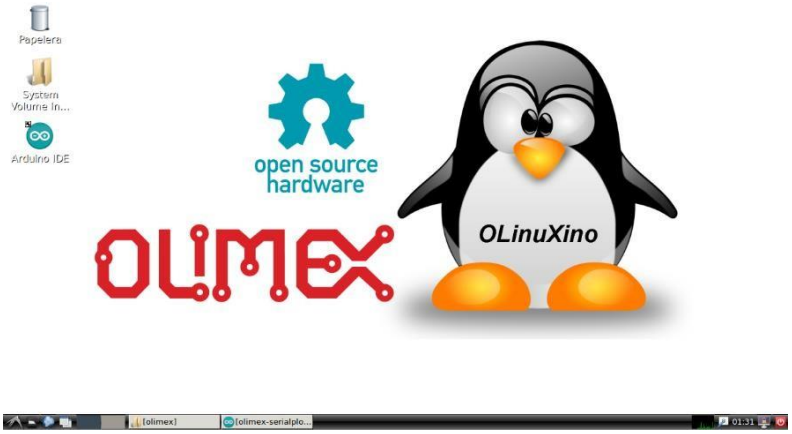


Figura 5: **Debian GNU/Linux en modo gráfico**

10) Se inicia sesión de la Olimex A20, se puede observar que es una distribución Debian **GNU/Linux**

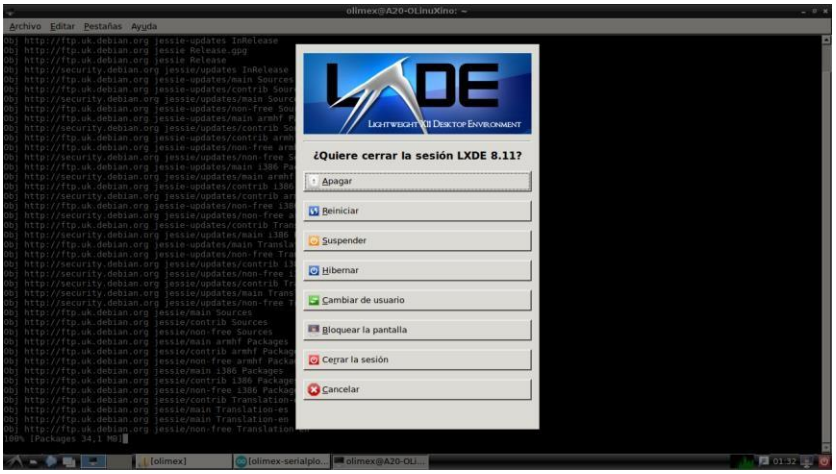


Figura 6: Distribución Debian **GNU/Linux**

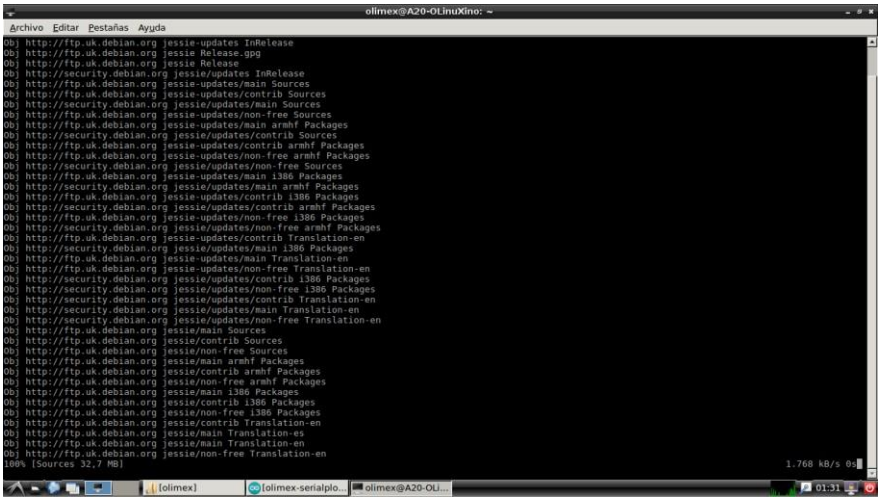


Figura 7: Lectura de paquetes

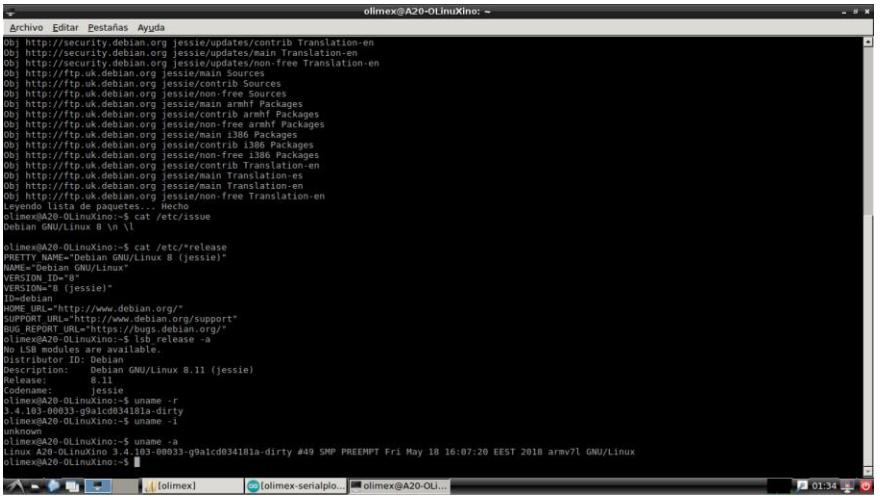


Figura 8: Ejecución de comandos uname

11) Cargar el código al ESPduino mediante el IDE de Arduino previamente descargado e instalado en la tarjeta Olimex A20.

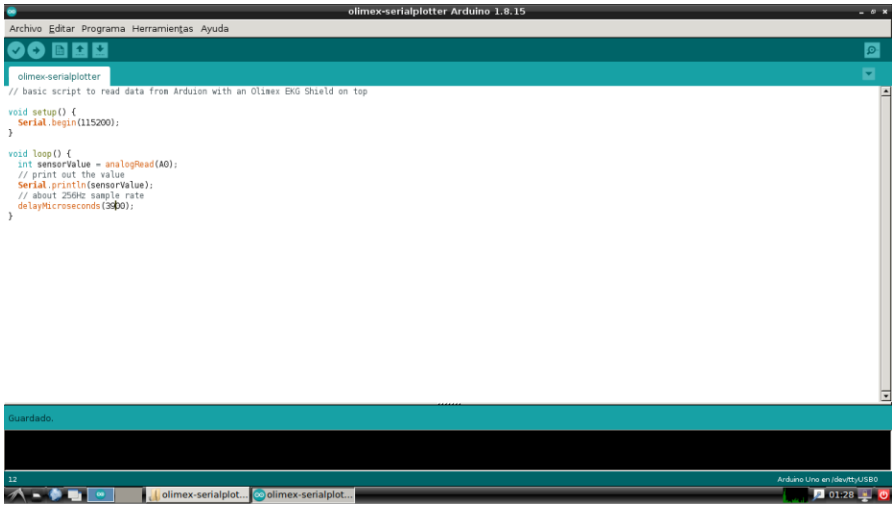


Figura 9: Ide de Arduino

12) Seleccionar monitor plotter para visualizar la lectura de la señal del electrocardiograma

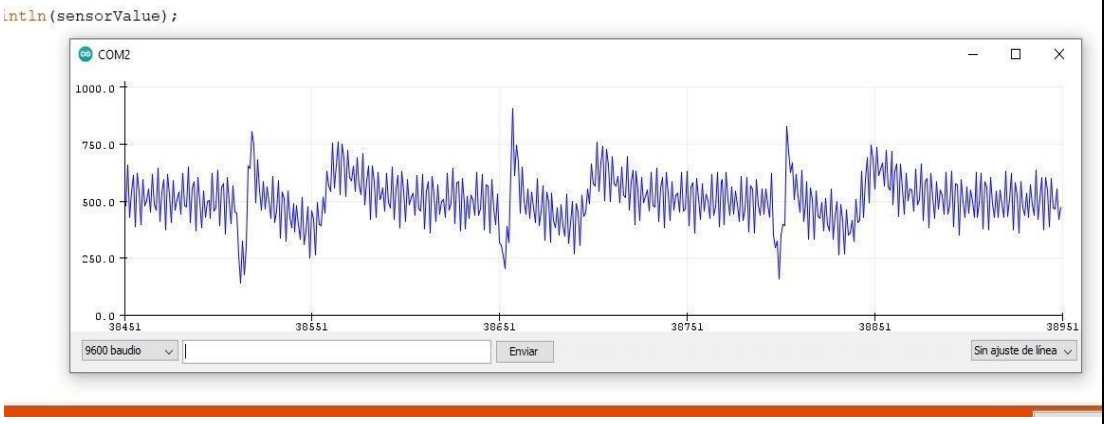


Figura 10: Lectura de Electrocardiograma

13) También se puede utilizar la placa con el cable serie / USB con conector UEXT, sin embargo, por temas de electrostática y diferentes tierras entra la laptop y la placa Olimex, no se podrá obtener una buena lectura.

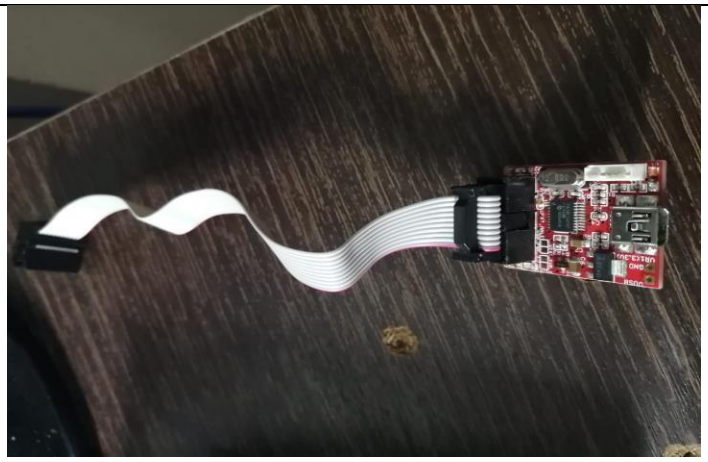


Figura 13: Convertidor serie / USB con conector UEXT

14) Para obtener las señales del electrocardiograma se utiliza los sensores L, R, D tipo pulsera

Figura 14: Cables SHIELD-EKG-EMG-PA para capturar señales

Código para subir EKG a Udidots

```
//UNIVERSIDAD POLITÉCNICA SALESIANA - SEDE GUAYAQUIL
//INGENIERÍA EN ELECTRÓNICA
//PROTOTIPO DE TELEMEDICINA - IOT
//AUTORES: MIGUEL ORTIZ - CRISTIAN MACIAS

// EKG - UDIDOTS

//LIBRERIAS
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiUdp.h>
#include <PubSubClient.h>
#include <NTPClient.h>

//DEFINIR VARIABLES
#define WIFISSID "Telemedicina" // Enter WifiSSID here
#define PASSWORD "Telemedicina.2021*" // Enter password here
#define TOKEN "BBFF-LvNOJ4z5bmM2UESky89WDoqT5OmLCr" // Ubidots' TOKEN
#define MQTT_CLIENT_NAME "mymqttclient" // MQTT client Name
#define VARIABLE_LABEL "ekg_olimex" // ubidots variable label
#define DEVICE_LABEL "ekg_olimex" // ubidots device label

const int analogInPin = 39; // Output Olimex is connected to GPIO35

char mqttBroker[] = "industrial.api.ubidots.com";
char payload[10000];
char topic[150];
```

```
// Space to store values to send
char str_sensor[10];
char str_millis[20];
double epochseconds = 0;
double epochmilliseconds = 0;
double current_millis = 0;
double current_millis_at_sensordata = 0;
double timestampp = 0;
int j = 0;

WiFiClient ubidots;
PubSubClient client(ubidots);
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");

/****************************************
   Auxiliar Functions
 ****************************************/
void callback(char* topic, byte* payload, unsigned int length) {
  char p[length + 1];
  memcpy(p, payload, length);
  p[length] = NULL;
  Serial.write(payload, length);
  Serial.println(topic);
}
void reconnect() {
  // Loop until we're reconnected
```

```
  while (!client.connected()) {
    Serial.println("Attempting MQTT connection...");
    // Attemp to connect
    if (client.connect(MQTT_CLIENT_NAME, TOKEN, "")) {
      Serial.println("Connected");
    } else  {
      Serial.print("Failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 2 seconds");
      // Wait 2 seconds before retrying
      delay(2000);
    }
  }
}

/**************************************
   Main Functions
 **************************************/
void setup() {
  Serial.begin(115200);
  analogReadResolution(10);
  WiFi.begin(WIFISSID, PASSWORD);
  Serial.println();
  Serial.print("Waiting for WiFi...");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
```

```
  Serial.println("");
  Serial.println("WiFi Connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  timeClient.begin();
  client.setServer(mqttBroker, 1883);
  client.setCallback(callback);
  timeClient.update();
  epochseconds = timeClient.getEpochTime();
  epochmilliseconds = epochseconds * 1000;
  Serial.print("epochmilliseconds=");
  Serial.println(epochmilliseconds);
  current_millis = millis();
  Serial.print("current_millis=");
  Serial.println(current_millis);
}

void loop() {
  if (!client.connected()) {
    reconnect();
    j = 0;
  }

  j = j + 1;
  // Serial.print("j=");
  // Serial.println(j);
  sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
  sprintf(payload, "%s", ""); // Cleans the payload
```

```
        sprintf(payload, "{\"%s\": [", VARIABLE_LABEL); // Adds the variable label

        float sensor = analogRead(analogInPin);
        dtostrf(float(sensor), 4, 2, str_sensor);
        Serial.print("Value: ");
        Serial.println(sensor);

        current_millis_at_sensordata = millis();
        timestampp = epochmilliseconds + (current_millis_at_sensordata - current_millis);
        dtostrf(timestampp, 10, 0, str_millis);
        sprintf(payload, "%s{\"value\":%s, \"timestamp\": %s}]}", payload, str_sensor, str_millis);
        // Serial.println("Publishing data to Ubidots Cloud");
        // Serial.println(payload);
        client.publish(topic, payload);
        delay(10);
      }
```

RESULTADO(S) OBTENIDO(S):

El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y descripciones de cada imagen

CONCLUSIONES:

El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

RÚBRICA DE REVISIÓN DE PRÁCTICA:

Fecha de realización de la práctica:	
Integrantes:	

	Malo	**Regular**	**Bueno**	**Excelente**	**Observaciones**
Sustentación correcta de las prácticas 35% del puntaje					
Eficacia, evidencia ilustraciones y correcta organización del progreso de la práctica 35% del puntaje					
Des enlaces de					

	las prácticas 30% del puntaje					
					PUNTAJE:	**/10**

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR		**GUÍA DE PRÁCTICA DE LABORATORIO**
CARRERA: INGENIERÍA ELECTRÓNICA		**ASIGNATURA**:
NRO. PRÁCTICA:		**TÍTULO PRÁCTICA**: Diseño e implementación de banco de prueba de Telemedicina con electrocardiograma de 3 vías EKG, ESP32 y AD8232.

OBJETIVO GENERAL:

Aprender el uso del banco de pruebas de telemedicina con el electrocardiograma de 3 vías EKG, ESP32 y AD8232

OBJETIVOS ESPECÍFICOS:

- Realizar configuración de EKG con ESP32.
- Visualizar datos desde IDE de Arduino.
- Visualizar datos del EKG en Ubidots.
- Almacenamiento de datos en Amazon Bucket S3

INSTRUCCIONES	1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo.
	2. Los estudiantes deben utilizar los equipos del banco de pruebas de telemedicina de una manera responsable y calificada para evitar daños en los equipos.
	3. Los estudiantes deben trabajar en grupo para el desarrollo de la práctica.
	4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta.

ACTIVIDADES POR DESARROLLAR:

Conexiones de hardware:

- Para esta práctica se usará el módulo de Telemedicina que contiene la placa AD8232 embebida en el prototipo IoT, este módulo se alimenta mediante el cable USB de la Laptop o con un cargador de 5 V 2A.

- Para las mediciones es importante tomar en cuenta que el paciente debe estar recostado y no debe tener elementos metálicos en su cuerpo como reloj, correa, pulseras, collares, modernas, etc.

- Para una adecuada medición el paciente debe estar relajado y tranquilo, recostado en un sillón.

- Tanto hombre como mujeres deben quitarse la ropa superior para que no haya interferencias en las mediciones.

- Las pruebas duran entre 2 a 3 minutos, para este caso donde es un prototipo educativo podría durar más tiempo.

- Es recomendable que también se retire la carga de la laptop para evitar interferencias en el aterrizaje de la red eléctrica.

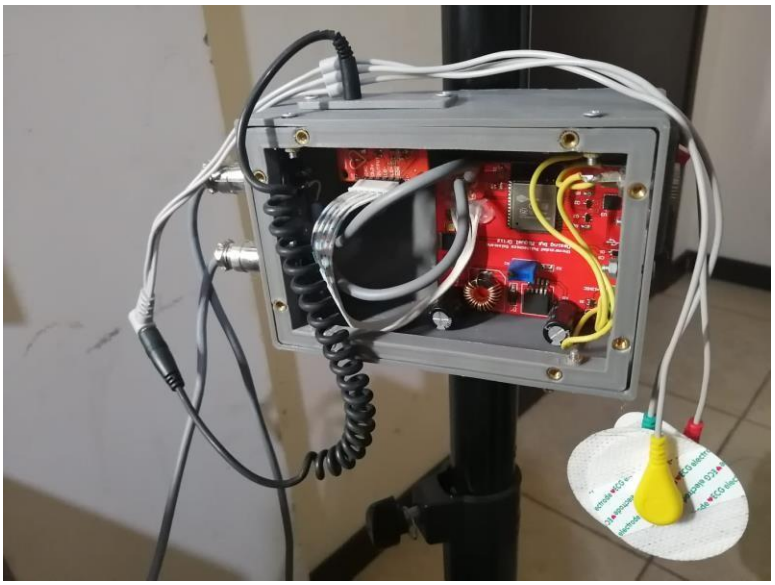


Figura 1: Prototipo de Telemedicina IoT

1) Para las mediciones se utilizará los electrodos de 3 vías mostrados en la figura 2.

Figura 2: Cables y electrodos

2) Se dispone de electrodos profesionales marca 3M para todas las pruebas con los EKG.

Figura 3: Electrodos 3M

3) La posición correcta de los electrodos se muestra en la siguiente figura**.** Para estas pruebas se utiliza la opción de electrodos en el pecho y costado.

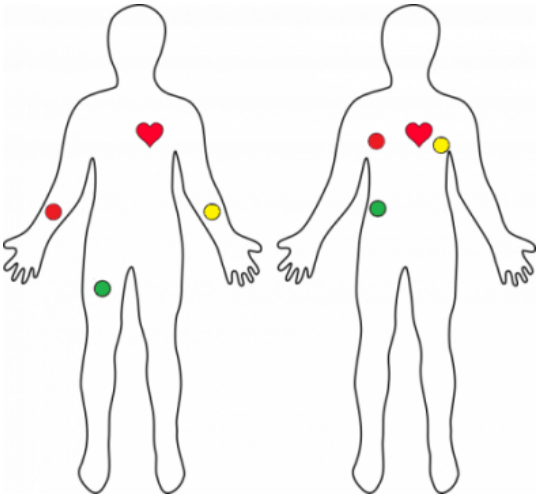


Figura 4: Colocación de electrodos

4) Se colocan los electrodos de acuerdo con la figura anterior, en pecho y abdomen.

Figura 5: Colocación de electrodos

Configuración de Software.

5) Para la toma de datos con el prototipo se realiza las conexiones adecuadas para el correcto funcionamiento del prototipo.

- La alimentación del prototipo de telemedicina es mediante cable USB.

- Una vez conectado el cable USB se debe validar que se reconozca el puerto en la PC.

- Para este caso el puerto configurado es el COM12

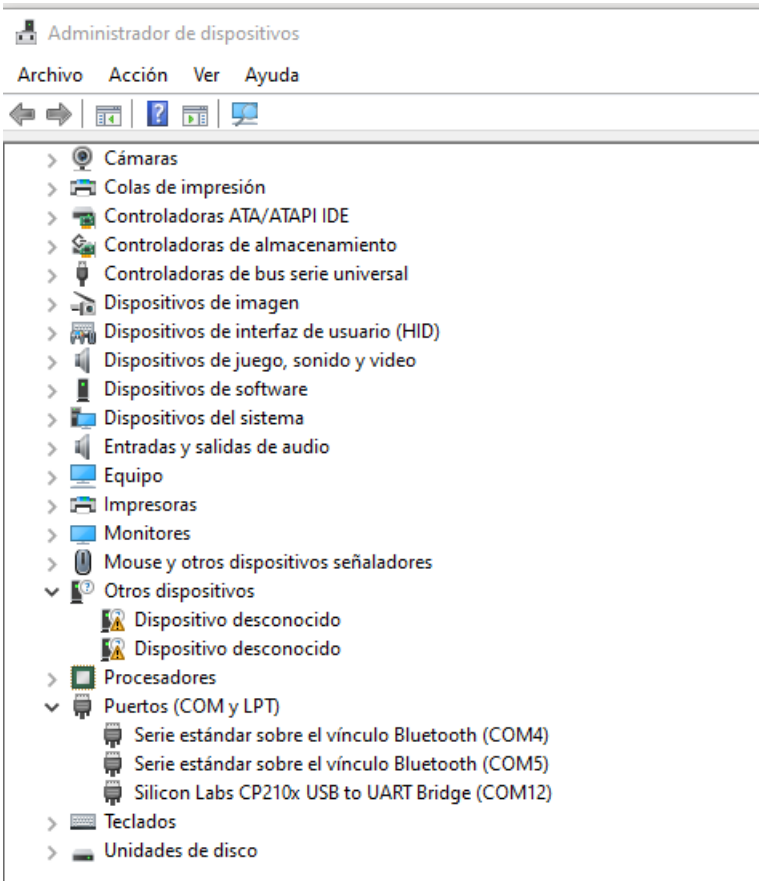


Figura 6: Puerto COM reconocido

6) Una vez reconocido el puerto, se debe realizar la configuración en el software IDE de Arduino.

7) Se instala la última versión del software el cual es el 1.8.15

Figura 7: Versión de IDE Arduino

Configuración en IDE Arduino:

8) Previo a la carga del código a Arduino se debe validar que el IDE Arduino esté correctamente configurado para la carga de las librerías y la placa correcta.

En preferencias se debe validar que el URL sea:
https://dl.espressif.com/dl/package_esp32_index.json

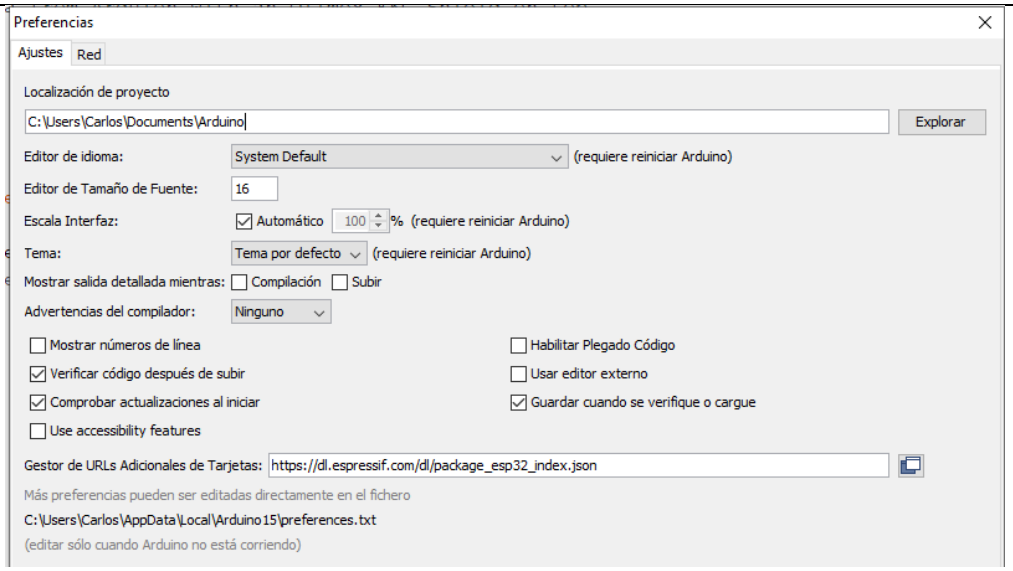


Figura 8: Preferencias de Arduino

9) En la opción de herramientas se debe descargar el módulo del ESP32 en la opción de gestor de tarjetas, la última versión es la 1.0.6.

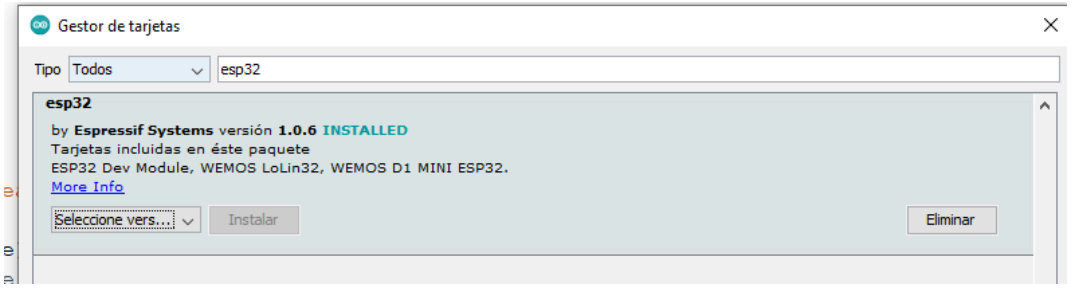


Figura 9: ESP32 gestor de tarjetas

10) Una vez instalada la tarjeta en el IDE de Arduino, se procede con la selección de la tarjeta ESP32 Dev module previo a la carga del código.

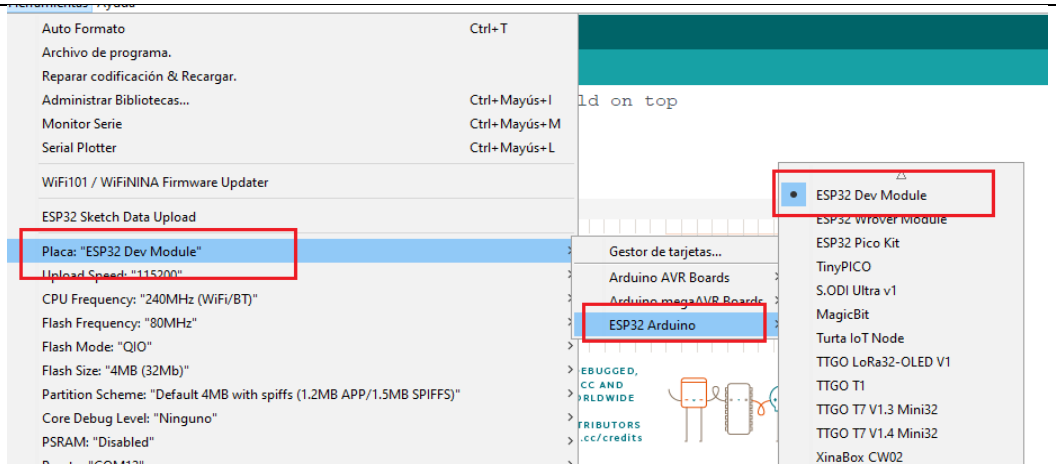


Figura 10: Selección de ESP32

11) Las configuraciones previo a la carga son las que se muestran en la figura 11, el puerto COM para este caso fue el COM12.

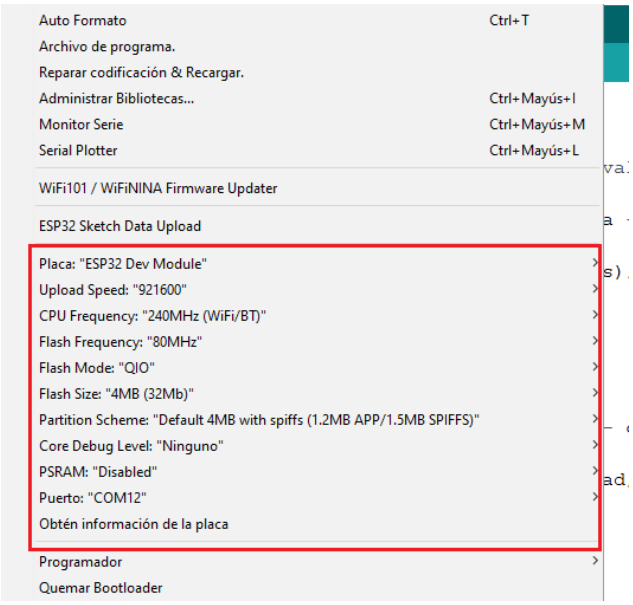


Figura 11: Configuraciones previo a la carga del código.

12) A continuación, se muestra el código que se carga en el módulo ESP32 del prototipo IoT de Telemedicina.

Hay que considerar que si aparece algún error se debe validar que la librerías esté correctamente instalada.

Las librerías que se debe cargar previamente son:

```
#include <WiFi.h>
#include <WiFiUdp.h>
#include <PubSubClient.h>
#include <NTPClient.h>
```

El WIFI es el SSID y Password del router utilizado en la práctica.

El Token ID de Udidots se lo obtiene de la cuenta Udidots previamente creada para las pruebas.

Código de ESP32 del EKG AD8232

```
//UNIVERSIDAD POLITÉCNICA SALESIANA - SEDE GUAYAQUIL
//INGENIERÍA EN ELECTRÓNICA
//PROTOTIPO DE TELEMEDICINA - IOT
//AUTORES: MIGUEL ORTIZ - CRISTIAN MACIAS

// EKG - UDIDOTS
```

```
//LIBRERIAS
#include <WiFi.h>
#include <WiFiUdp.h>
#include <PubSubClient.h>
#include <NTPClient.h>

//DEFINIR VARIABLES
#define WIFISSID "Telemedicina" // Enter WifiSSID here
#define PASSWORD "Telemedicina.2021*" // Enter password here
#define TOKEN "BBFF-LvNOJ4z5bmM2UESky89WDoqT5OmLCr" // Ubidots' TOKEN
#define MQTT_CLIENT_NAME "mymqttclient" // MQTT client Name
#define VARIABLE_LABEL "ekg" // ubidots variable label
#define DEVICE_LABEL "esp32-telemedicina" // ubidots device label

#define SENSORPIN A6 // Set the A0 as SENSORPIN

char mqttBroker[]  = "industrial.api.ubidots.com";
char payload[10000];
char topic[150];

// Space to store values to send
char str_sensor[10];
char str_millis[20];
double epochseconds = 0;
double epochmilliseconds = 0;
double current_millis = 0;
double current_millis_at_sensordata = 0;
```

```
double timestampp = 0;
int j = 0;

/**************************************
  Auxiliar Functions
 **************************************/

WiFiClient ubidots;
PubSubClient client(ubidots);
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");



void callback(char* topic, byte* payload, unsigned int length) {
  char p[length + 1];
  memcpy(p, payload, length);
  p[length] = NULL;
  Serial.write(payload, length);
  Serial.println(topic);
}
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.println("Attempting MQTT connection...");
    // Attemp to connect
    if (client.connect(MQTT_CLIENT_NAME, TOKEN, "")) {
      Serial.println("Connected");
    } else {
```

```
      Serial.print("Failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 2 seconds");
      // Wait 2 seconds before retrying
      delay(2000);
    }
  }
}
/**************************************
  Main Functions
 **************************************/
void setup() {
  Serial.begin(115200);
  analogReadResolution(10);
  WiFi.begin(WIFISSID, PASSWORD);
  pinMode(SENSORPIN, INPUT);
  Serial.println();
  Serial.print("Waiting for WiFi...");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.println("WiFi Connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  timeClient.begin();
  client.setServer(mqttBroker, 1883);
```

```
  client.setCallback(callback);
  timeClient.update();
  epochseconds = timeClient.getEpochTime();
  epochmilliseconds = epochseconds * 1000;
  Serial.print("epochmilliseconds=");
  Serial.println(epochmilliseconds);
  current_millis = millis();
  Serial.print("current_millis=");
  Serial.println(current_millis);
 }
 void loop() {
  if (!client.connected()) {
   reconnect();
   j = 0;
  }

  j = j + 1;
  //Serial.print("j=");
  //Serial.println(j);
  sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
  sprintf(payload, "%s", ""); // Cleans the payload
  sprintf(payload, "{\"%s\": [", VARIABLE_LABEL); // Adds the variable
label

   float sensor = analogRead(SENSORPIN);
   dtostrf(float(sensor), 4, 2, str_sensor);
   Serial.print("Value: ");
   Serial.println(sensor);
```

```
            current_millis_at_sensordata = millis();
            timestampp = epochmilliseconds + (current_millis_at_sensordata -
    current_millis);
            dtostrf(timestampp, 10, 0, str_millis);
            sprintf(payload, "%s{\"value\":%s, \"timestamp\": %s}]}", payload,
    str_sensor, str_millis);
          // Serial.println("Publishing data to Ubidots Cloud");
          // Serial.println(payload);
          client.publish(topic, payload);
          delay(20);
    }
```

Luego de cargar el código se debe validar que no haya algún error o problema con la compilación de la carga.

Si no aparece problemas se procede a abrir el Serial plotter para visualizar localmente la data obtenida del AD8232.

Serial Plotter:

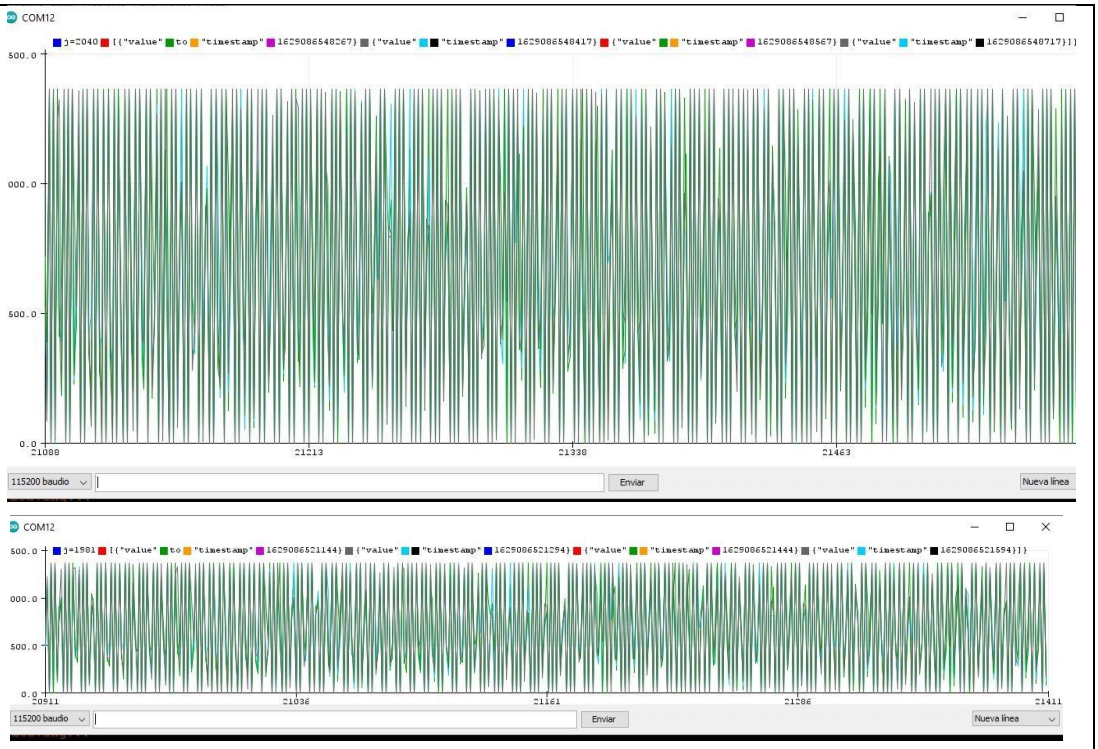


Figura 12: Serial Plotter del EKG.

Para esta práctica donde los datos del EKG se visualizan localmente en una PC y en servidores en la web, se elige trabajar con Ubidots que es una herramienta en la nube para la visualización de los datos del EKG en tiempo real.

Hay que considerar que los datos subidos a los servidores de Ubidots presentarán una latencia por consecuencia de la red y los saltos que tenga que dar los paquetes de datos para mostrar la data. Sin embargo, el código logra tener buenos resultados en cuanto a visualización de información.

Ubidots:

Se crea una cuenta en udidots para las pruebas del prototipo. Ubidots tiene una opción de cuenta gratuita llamada STEM de Ubidots, el cual tiene límite de configuración de 1 dispositivo máximo y de recibir o transmitir máximo 4000 datos al día.

Para el caso de las pruebas del EKG esta cuenta no era suficiente, por tal motivo se opta por comprar una cuenta IoT Entrepeneur.

https://ubidots.com/pricing/

El costo de la cuenta es de $49 al mes.

Con esta cuenta se tiene las siguientes ventajas: 25 variables por dispositivo y 2 millones de datos por mes. El cual es suficiente para las pruebas del prototipo.

Se recomienda que para las pruebas de laboratorio con estudiantes se utilice la cuenta gratuita STEM de Udidots, sabiendo la limitante de 4000 dots por dias.

Se ingresa con user y pass a la cuenta de Ubidots.

Se configura el dispositivo en la sección devices. El nombre es Telemedicina.

Para agregar el dispositivo se da click en Add filter – Blank Devices.

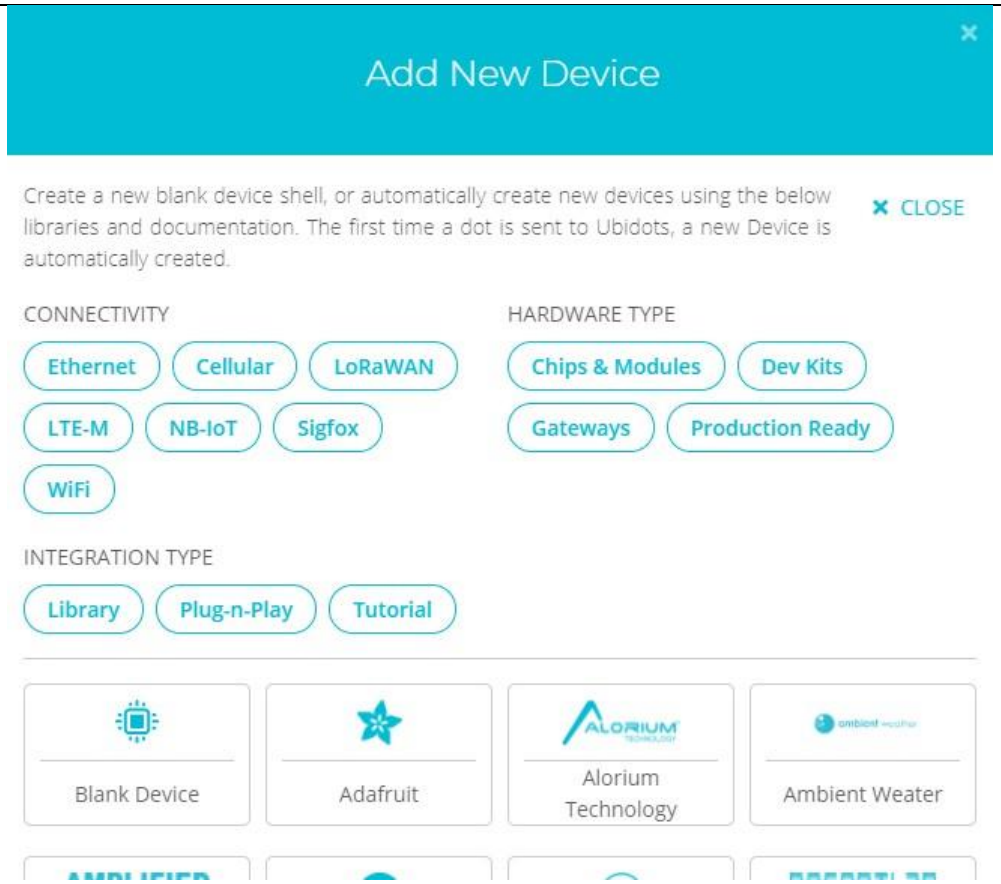


Figura 13: Agregar dispositivo

Se coloca el nombre del dispositivo y el nombre de la capa.

- Name: Telemedicina
- API Label: esp32-telemedicina

Estos datos son importantes ya que se colocan también en el código que se quema en el ESP32.

Cuando se crea el dispositivo se crea automáticamente el TOKEN

BBFF-Il42N2R4UdsOGXPsWyYGCc8upD9hl4

Esto varía dependiendo de la cuenta de udidots.

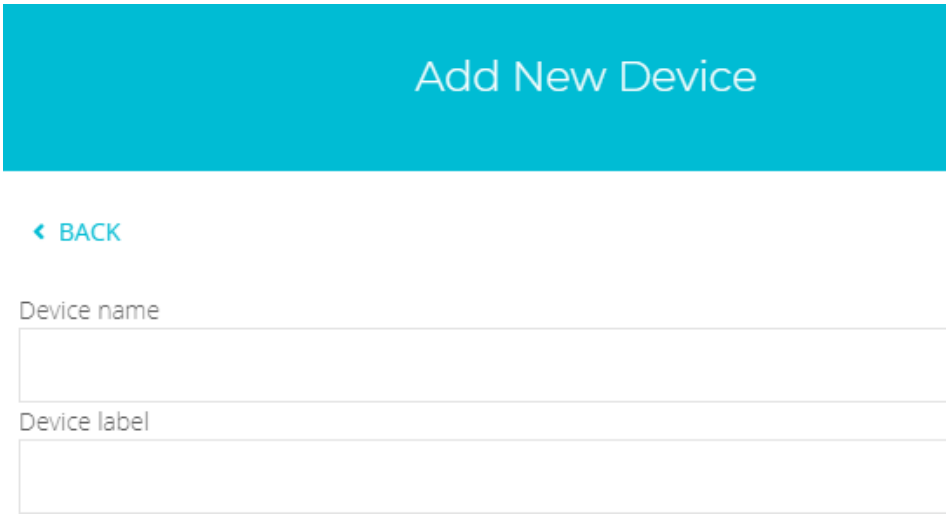


Figura 14: Agregar dispositivo

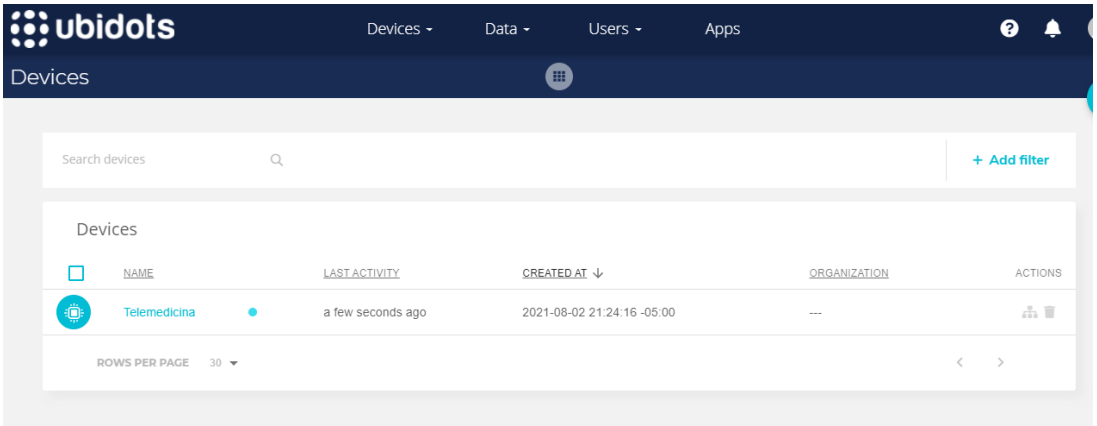


Figura 15: Serial Plotter del EKG.

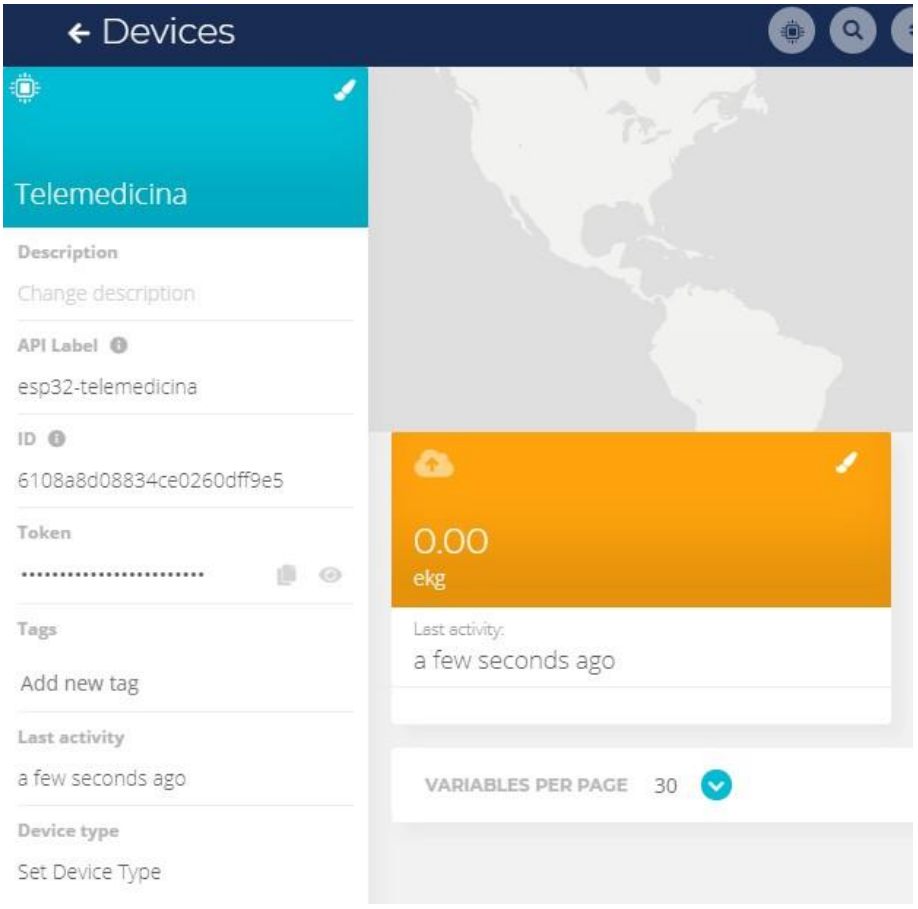


Figura 16: Configuración de Devices.

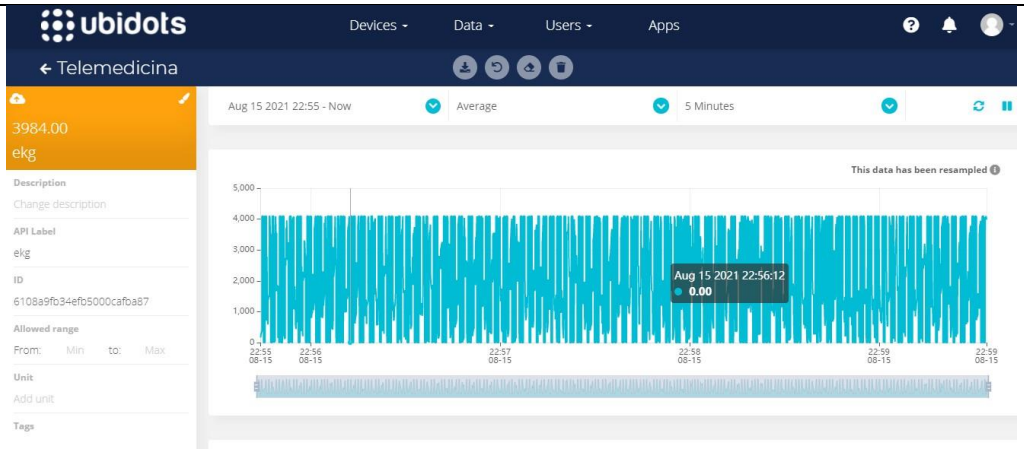


Figura 17: Datos obtenidos por el dispositivo.

Para crear las gráficas se debe dar click en Data – Dashboards.

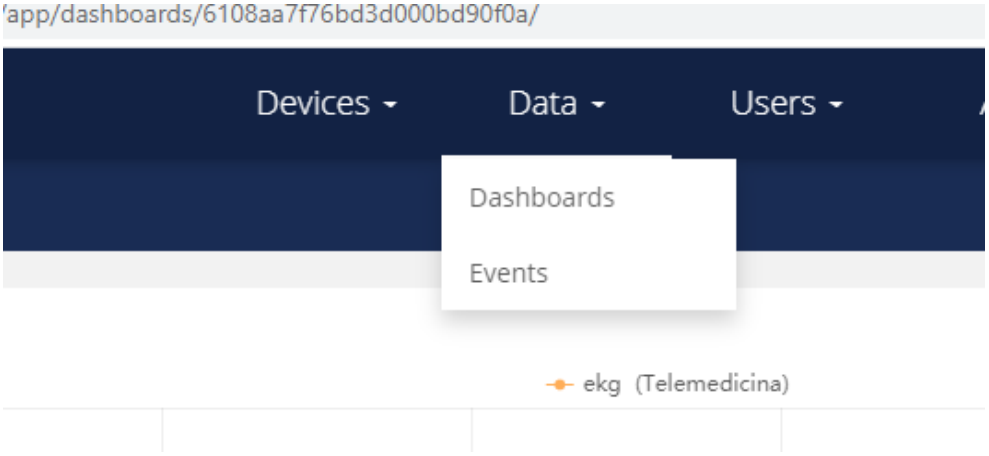


Figura 18: Agregar dashboard.

Dar click en Add New Widget y elegir line chart.

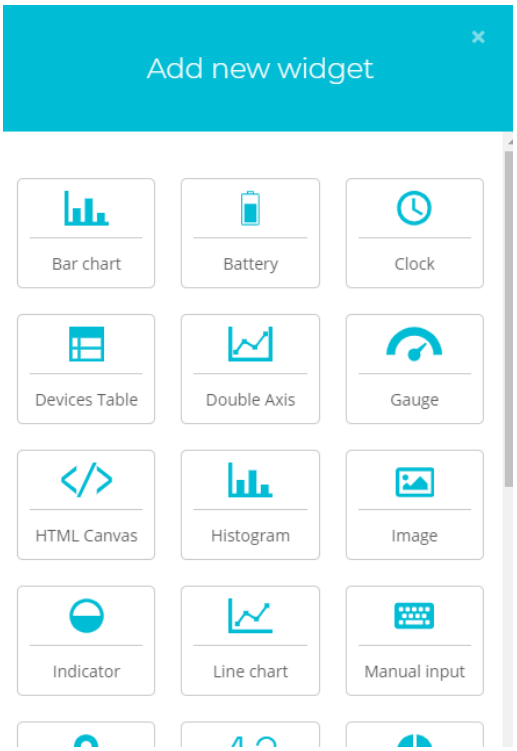


Figura 19: Agregar line chart

A continuación se muestra la configuración de Line Chart.

El nombre ekg es el mismo que se encuentra en el código cargado en el ESP32.

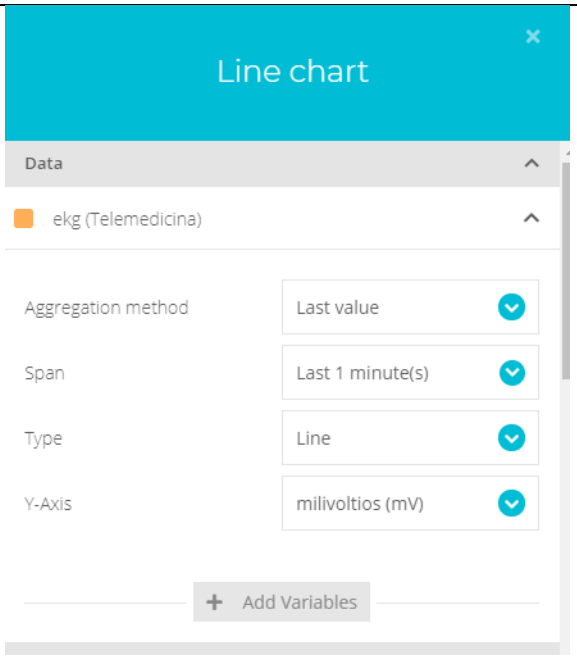


Figura 20: Configuración Line Chart

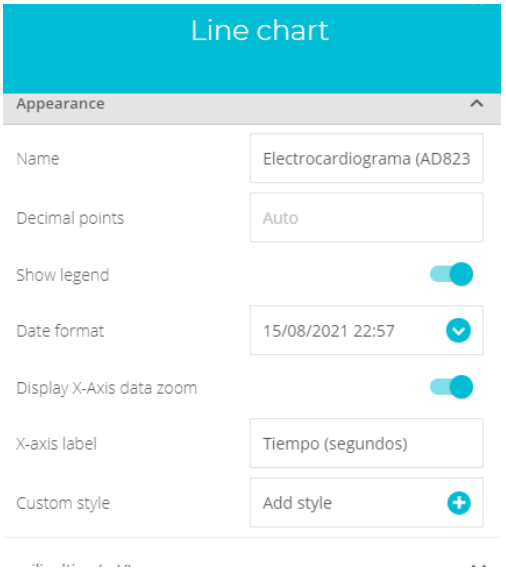


Figura 21: Configuración Line Chart – Formato hora

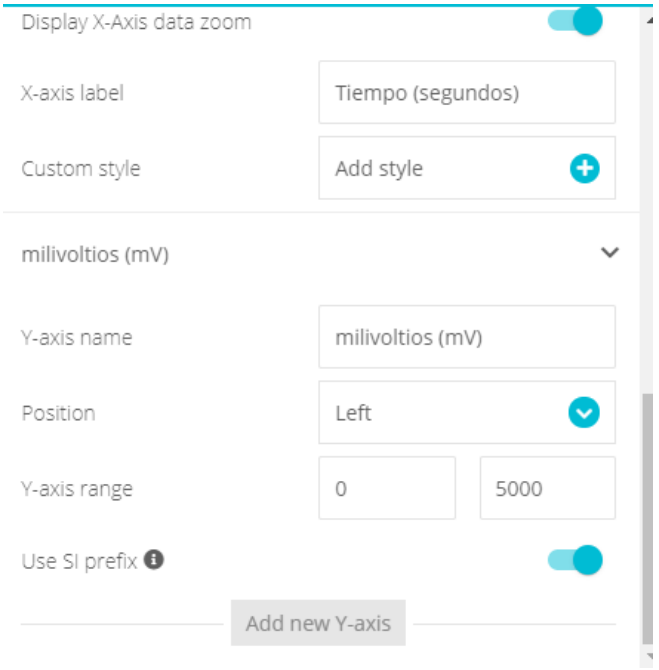


Figura 22: Configuración Line Chart – Axis X - Y

Figura 23: Visualización de datos del EKG en Ubidots

Los datos obtenidos pueden ser descargados desde la plataforma Ubidots. En la sección de Data Export en la cuenta, se debe seleccionar Export y la data será enviada al mail configurado en la cuenta de Ubidots.

RESULTADO(S) OBTENIDO(S):

El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y descripciones de cada imagen.

CONCLUSIONES:

El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

RÚBRICA DE REVISIÓN DE PRÁCTICA:

Fecha de realización de la práctica:	
Integrates:	

	Malo	**Regular**	**Bueno**	**Excelente**	**Observaciones**
Sustentación correcta de las prácticas 35% del puntaje					
Eficacia, evidencia ilustraciones y correcta organización del progreso de la práctica 35% del					

	puntaje					
	Des enlaces de las prácticas 30% del puntaje					
					PUNTAJE:	**/10**

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR		**GUÍA DE PRÁCTICA DE LABORATORIO**
CARRERA: INGENIERÍA ELECTRÓNICA		**ASIGNATURA**:
NRO. PRÁCTICA:		**TÍTULO PRÁCTICA**: Diseño e implementación de banco de prueba de telemedicina con báscula digital inalámbrica con conexión al ESP32.

OBJETIVO GENERAL:

Aprender el uso del banco de pruebas de Telemedicina con báscula digital inalámbrica con conexión al ESP32.

OBJETIVOS ESPECÍFICOS:

- Conectorizar báscula digital al ESP32.
- Monitorizar datos de báscula digital.
- Enviar información de báscula digital a Ubidots.

INSTRUCCIONES	1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo.
	2. Los estudiantes deben utilizar los equipos del banco de pruebas de telemedicina de una manera responsable y calificada para evitar daños en los equipos.
	3. Los estudiantes deben trabajar en grupo para el desarrollo de la práctica.
	4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta.

ACTIVIDADES POR DESARROLLAR:

Para esta práctica se utiliza la báscula digital inalámbrica Xiaomi MI Smart Scale 2 el cual se conectará por bluetooth al ESP32 y posteriormente se enviará el valor del peso en kilogramos a Ubidots.

Figura 1: Báscula digital inalámbrica Xiaomi MI Smart Scale 2

1) En primer lugar, se debe obtener la mac address del bluetooth de la balanza, para esto se debe descargar en el celular desde Google Play la aplicación Mi Fit de Xiaomi.

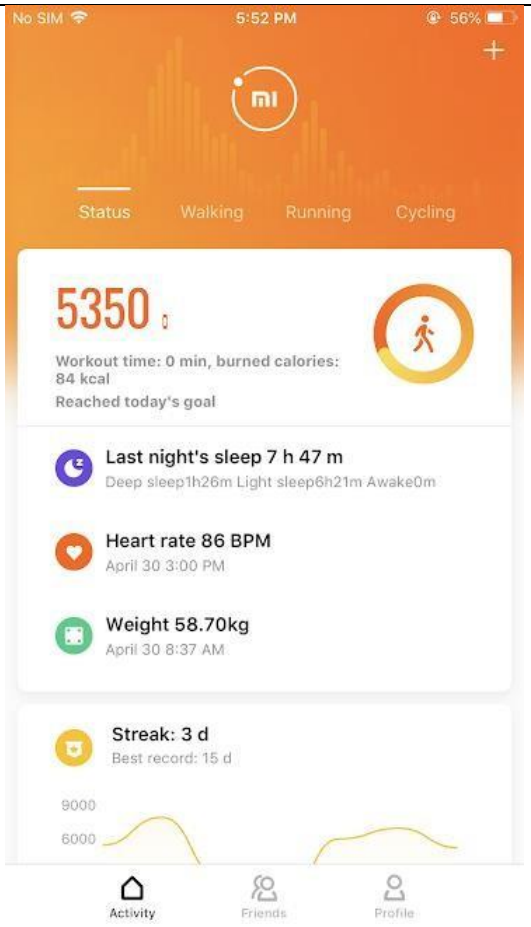


Figura 2: APP Mi Fit.

2) Para visualizar la mac address del bluetooth se debe ir a la opción de configuración y visualizar la mac address.

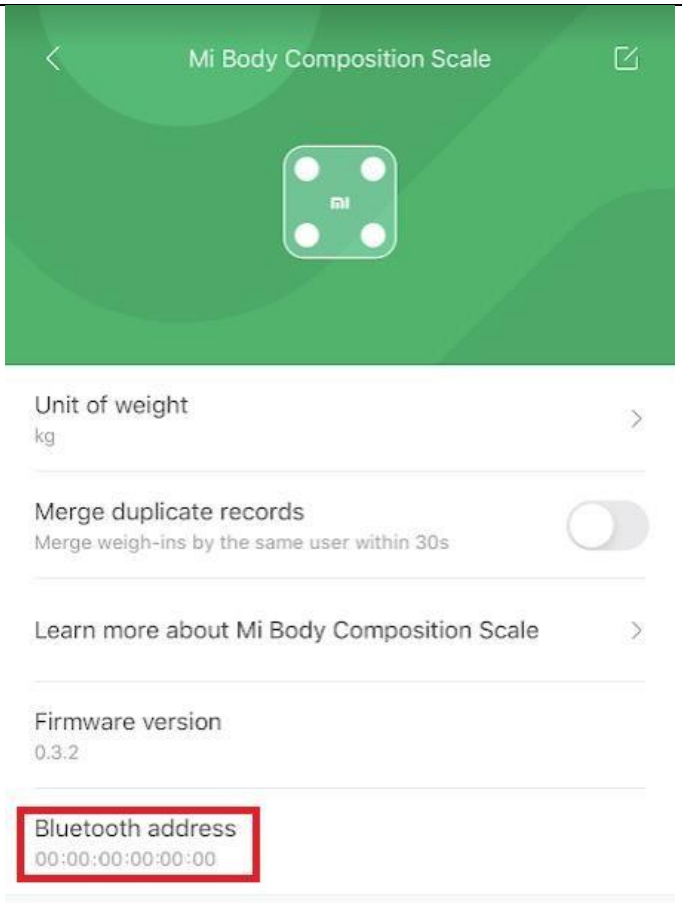


Figura 3: Mac address bluetooth de la balanza inalámbrica.

Se realiza la carga del siguiente código en el cual se coloca la mac de la balanza.

Código:

```
#include <Arduino.h>
#include <BLEDevice.h>
//#include "BLEScan.h"
```

```
// The remote service we wish to connect to.
static BLEUUID serviceUUID("0000181b-0000-1000-8000-00805f9b34fb");
// The characteristic of the remote service we are interested in.
static BLEUUID    charUUID("00002a9c-0000-1000-8000-00805f9b34fb");

static boolean doConnect = false;
static boolean connected = false;
static boolean doScan = false;
static BLERemoteCharacteristic* pRemoteCharacteristic;
static BLEAdvertisedDevice* myDevice;

static void notifyCallback(
  BLERemoteCharacteristic* pBLERemoteCharacteristic,
  uint8_t* pData,
  size_t length,
  bool isNotify) {
    //Console debugging
    Serial.print("Received data. Length = ");
    Serial.print(length);
    Serial.print(". - Data bytes: ");
    for(int i =0; i< length; i++){
    Serial.print(pData[i]);
    Serial.print("  ");
    }
    Serial.println(" ");
    //End of console debugging

    double weight = 0;
```

```
    weight = (pData[11] + pData[12] * 256) * 0.005;

    Serial.print("Weight: ");
    Serial.print(weight);
    Serial.println(" Kg - ");
}

class MyClientCallback : public BLEClientCallbacks {
  void onConnect(BLEClient* pclient) {
  }

  void onDisconnect(BLEClient* pclient) {
    connected = false;
    Serial.println("onDisconnect");
  }
};

bool connectToServer() {
    Serial.print("Forming a connection to ");
    Serial.println(myDevice->getAddress().toString().c_str());

    BLEClient*  pClient  = BLEDevice::createClient();
    Serial.println(" - Created client");

    pClient->setClientCallbacks(new MyClientCallback());

    // Connect to the remove BLE Server.
    pClient->connect(myDevice);  // if you pass BLEAdvertisedDevice instead
```

```
of address, it will be recognized type of peer device address (public or private)
        Serial.println(" - Connected to server");

        // Obtain a reference to the service we are after in the remote BLE server.
        BLERemoteService* pRemoteService = pClient->getService(serviceUUID);
        if (pRemoteService == nullptr) {
          Serial.print("Failed to find our service UUID: ");
          Serial.println(serviceUUID.toString().c_str());
          pClient->disconnect();
          return false;
        }
        Serial.println(" - Found our service");


        // Obtain a reference to the characteristic in the service of the remote BLE server.
        pRemoteCharacteristic = pRemoteService->getCharacteristic(charUUID);
        if (pRemoteCharacteristic == nullptr) {
          Serial.print("Failed to find our characteristic UUID: ");
          Serial.println(charUUID.toString().c_str());
          pClient->disconnect();
          return false;
        }
        Serial.println(" - Found our characteristic");

        // Read the value of the characteristic.
        if(pRemoteCharacteristic->canRead()) {
```

```
            std::string value = pRemoteCharacteristic->readValue();
            Serial.print("The characteristic value was: ");
            Serial.println(value.c_str());
          }

          //if(pRemoteCharacteristic->canNotify())
          pRemoteCharacteristic->registerForNotify(notifyCallback);

          connected = true;
          return true;
        }
        /**
         * Scan for BLE servers and find the first one that advertises the service we are looking for.
         */
        class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {
         /**
           * Called for each advertising BLE server.
           */
          void onResult(BLEAdvertisedDevice advertisedDevice) {
            Serial.print("BLE Advertised Device found: ");
            Serial.println(advertisedDevice.toString().c_str());

            // We have found a device, let us now see if it contains the service we are looking for.
            if (advertisedDevice.haveServiceUUID() && advertisedDevice.isAdvertisingService(serviceUUID)) {
```

```
            BLEDevice::getScan()->stop();
            myDevice = new BLEAdvertisedDevice(advertisedDevice);
            doConnect = true;
            doScan = true;

          } // Found our server
        } // onResult
      }; // MyAdvertisedDeviceCallbacks


      void setup() {
        Serial.begin(115200);
        Serial.println("Starting Arduino BLE Client application...");
        BLEDevice::init("");

        // Retrieve a Scanner and set the callback we want to use to be informed when we
        // have detected a new device.  Specify that we want active scanning and start the
        // scan to run for 5 seconds.
        BLEScan* pBLEScan = BLEDevice::getScan();
        pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());
        pBLEScan->setInterval(1349);
        pBLEScan->setWindow(449);
        pBLEScan->setActiveScan(true);
        pBLEScan->start(5, false);
```

```
} // End of setup.


// This is the Arduino main loop function.
void loop() {

  // If the flag "doConnect" is true then we have scanned for and found the desired
  // BLE Server with which we wish to connect.  Now we connect to it. Once we are
  // connected we set the connected flag to be true.
  if (doConnect == true) {
    if (connectToServer()) {
      Serial.println("We are now connected to the BLE Server.");
    } else {
      Serial.println("We have failed to connect to the server; there is nothin more we will do.");
    }
    doConnect = false;
  }

  // If we are connected to a peer BLE Server, update the characteristic each time we are reached
  // with the current time since boot.
  if (connected) {
    String newValue = "Time since boot: " + String(millis()/1000);
    //Serial.println("Setting new characteristic value to \"" + newValue + "\"");
```

```
            // Set the characteristic's value to be the array of bytes that is actually a
string.
            pRemoteCharacteristic->writeValue(newValue.c_str(),
newValue.length());
          }else if(doScan){
            BLEDevice::getScan()->start(0);  // this is just example to start scan after
disconnect, most likely there is better way to do it in arduino
          }

          delay(1000); // Delay a second between loops.
        } // End of Loop
```

3) Luego de la carga del código se puede visualizar los datos desde el monitor serie del IDE de Arduino.

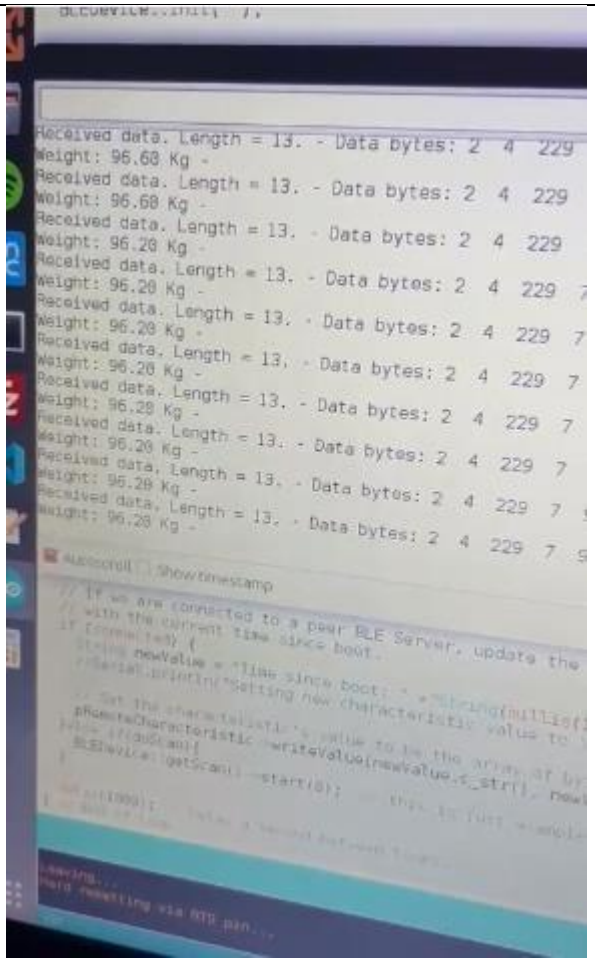


Figura 4. Datos de peso en Kg

4) Se envía los datos al servidor de Ubidots el cual se puede observar en el siguiente dashboard.

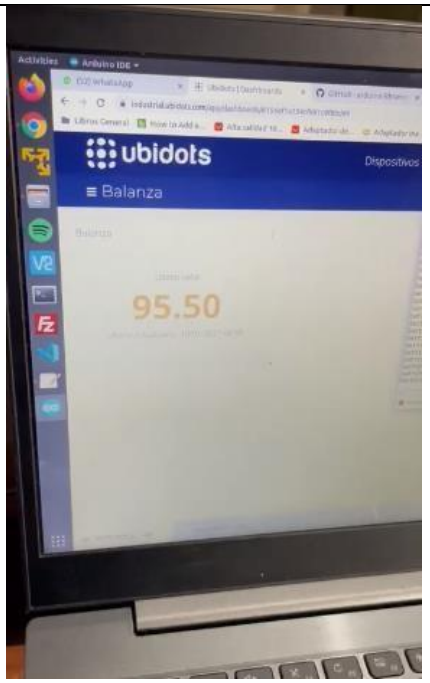


Fig. 5. Datos de peso en ubidots

RESULTADO(S) OBTENIDO(S):

El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y descripciones de cada imagen

CONCLUSIONES:

El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

RÚBRICA DE REVISIÓN DE PRÁCTICA:

Fecha de realización de la práctica:	
Integrantes:	

	Malo	**Regular**	**Bueno**	**Excelente**	**Observaciones**
Sustentación correcta de las prácticas 35% del puntaje					
Eficacia, evidencia ilustraciones y correcta organización del progreso de la práctica 35% del puntaje					
Des enlaces de					

	las prácticas 30% del puntaje					
					PUNTAJE:	**/10**

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR		**GUÍA DE PRÁCTICA DE LABORATORIO**
CARRERA: INGENIERÍA ELECTRÓNICA		**ASIGNATURA**:
NRO. PRÁCTICA:		**TÍTULO PRÁCTICA**: Diseño e implementación de banco de prueba de telemedicina con oxímetro digital con conexión al ESP32.

OBJETIVO GENERAL:

Aprender el uso del banco de pruebas de Telemedicina con oxímetro digital con conexión al Esp32.

OBJETIVOS ESPECÍFICOS:

- Realizar la configuración de oxímetro digital

<table>
<tr><td colspan="2">

- Visualizar la información del oxímetro en servidores web de Ubidots.
- Respaldar la información en Amazon S3

</td></tr>
<tr><td rowspan="4">INSTRUCCIONES</td><td>1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo.</td></tr>
<tr><td>2. Los estudiantes deben utilizar los equipos del banco de pruebas de telemedicina de una manera responsable y calificada para evitar daños en los equipos.</td></tr>
<tr><td>3. Los estudiantes deben trabajar en grupo para el desarrollo de la práctica.</td></tr>
<tr><td>4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta.</td></tr>
<tr><td colspan="2">

ACTIVIDADES POR DESARROLLAR:

1) Para el uso del oxímetro digital se carga el siguiente código en el ESP32.

Código ESP32 oxímetro digital

```
//UNIVERSIDAD POLITÉCNICA SALESIANA - SEDE GUAYAQUIL
//INGENIERÍA EN ELECTRÓNICA
//PROTOTIPO DE TELEMEDICINA - IOT
//AUTORES: MIGUEL ORTIZ - CRISTIAN MACIAS
```

</td></tr>
</table>

```
#include <Arduino.h>
#include <WiFi.h>
#include <Wire.h>
#include "Adafruit_GFX.h"
#include "Adafruit_SSD1306.h"
#include "MAX30100_PulseOximeter.h"
#include  "UbidotsEsp32Mqtt.h"
#include "Adafruit_MLX90614.h"

const char *UBIDOTS_TOKEN = "BBFF-LvNOJ4z5bmM2UESky89WDoqT5OmLCr";
const char *DEVICE_LABEL = "esp32-telemedicina";
const char *OXIGENO_LABEL = "oxigeno";
const char *BMP_LABEL = "heartrate";
const char *TEMPERATURA = "temperatura";

const long PUBLISH_FREQUENCY = 10000; // Update rate in milliseconds

const char *WIFI_SSID = "Telemedicina";      // Put here your Wi-Fi SSID
const char *WIFI_PASS = "Telemedicina.2021*";       // Put here your Wi-Fi password

long Ts_Last_Publish = 0;

float temperatura = 0.00;
float bpm = 0.0;
float spo2 = 0.0;
```

```
Ubidots ubidots(UBIDOTS_TOKEN);
Adafruit_MLX90614 mlx = Adafruit_MLX90614();

PulseOximeter pox;

 void callback(char *topic, byte *payload, unsigned int length){
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++){
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

void onBeatDetected(){
  // Serial.println("Beat!");
}

int scan_devices_i2c(){
  byte error, address;
  int nDevices;
  Serial.println("Scanning...");
  nDevices = 0;
  for(address = 1; address < 127; address++ ) {
    Wire.beginTransmission(address);
    error = Wire.endTransmission();
```

```
      if (error == 0) {
        // Serial.print("I2C device found at address 0x");
        // if (address<16) {
        //  Serial.print("0");
        // }
        Serial.println(address,HEX);
        nDevices++;
      }
      else if (error==4) {
        // Serial.print("Unknow error at address 0x");
        if (address<16) {
          // Serial.print("0");
        }
        Serial.println(address,HEX);
      }
    }
    return nDevices;

}

void setup(){
  Serial.begin(115200);
  delay(200);

  ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
  ubidots.setCallback(callback);
  ubidots.setup();
  ubidots.reconnect();
```

```
  Serial.println('\n');
  Serial.println("Connection established!");
  Serial.print("IP address:\t");
  Serial.println(WiFi.localIP());

  if (!pox.begin()){
    Serial.println("FAILED");
  }else{
    Serial.println("SUCCESS");
  }
  pox.setOnBeatDetectedCallback(onBeatDetected);
  mlx.begin();
}

void loop(){

  pox.update();

  if (millis() - Ts_Last_Publish > PUBLISH_FREQUENCY) {
    if(scan_devices_i2c() > 1){
      Serial.print("Turn-on Temp Gun");
      temperatura = mlx.readAmbientTempC();
    }else{
      Serial.print("Turn-off Temp Gun");
      temperatura = 0.00;
    }
```

```
      bpm = pox.getHeartRate();
      spo2 = pox.getSpO2();
      Serial.print("BPM: ");
      Serial.print(bpm);
      Serial.print("\t");
      Serial.print("SpO2: ");
      Serial.print(spo2);
      Serial.print("\t");
      Serial.print("Temperatura Corporal: ");
      Serial.println(temperatura);
      ubidots.add(BMP_LABEL, bpm);
      ubidots.add(OXIGENO_LABEL, spo2);
      ubidots.add(TEMPERATURA, temperatura);
      ubidots.publish(DEVICE_LABEL);
      Ts_Last_Publish = millis();
    }
  }
```

2) Una vez cargado el código se realizan pruebas con el oxímetro.

Figura 1: Oxímetro

3) Se puede visualizar los datos en el monitor del IDE de Arduino

4) Se observa en Udidots los datos obtenidos por el sensor pulsímetro.

RESULTADO(S) OBTENIDO(S):

El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y descripciones de cada imagen

CONCLUSIONES:

El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

RÚBRICA DE REVISIÓN DE PRÁCTICA:

Fecha de realización de la práctica:	
Integrantes:	

	Malo	**Regular**	**Bueno**	**Excelente**	**Observaciones**
Sustentación correcta de las prácticas 35% del puntaje					
Eficacia, evidencia ilustraciones y correcta organización del progreso de la práctica 35% del					

puntaje						
Des enlaces de las prácticas 30% del puntaje						
					PUNTAJE:	**/10**

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR		**GUÍA DE PRÁCTICA DE LABORATORIO**
CARRERA: INGENIERÍA ELECTRÓNICA		**ASIGNATURA**:
NRO. PRÁCTICA:		**TÍTULO PRÁCTICA**: Diseño e implementación de banco de prueba de Telemedicina con sensor de temperatura corporal y conectividad al ESP32.

OBJETIVO GENERAL:

Aprender el uso del banco de pruebas de telemedicina con sensor de

temperatura corporal y conectividad al ESP32.

OBJETIVOS ESPECÍFICOS:

- Realizar configuración de sensor de temperatura infrarrojos.
- Visualizar datos de temperatura en servicios de Ubidots.
- Tomar muestras del sensor de temperatura corporal.

INSTRUCCIONES	1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo.
	2. Los estudiantes deben utilizar los equipos del banco de pruebas de telemedicina de una manera responsable y calificada para evitar daños en los equipos.
	3. Los estudiantes deben trabajar en grupo para el desarrollo de la práctica.
	4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta.

ACTIVIDADES POR DESARROLLAR:

5) Para el uso del termómetro digital se carga el siguiente código en el ESP32.

Código ESP32 termómetro digital

```
//UNIVERSIDAD POLITÉCNICA SALESIANA - SEDE GUAYAQUIL
//INGENIERÍA EN ELECTRÓNICA
//PROTOTIPO DE TELEMEDICINA - IOT
//AUTORES: MIGUEL ORTIZ - CRISTIAN MACIAS

#include <Arduino.h>
#include <WiFi.h>
#include <Wire.h>
#include "Adafruit_GFX.h"
#include "Adafruit_SSD1306.h"
#include "MAX30100_PulseOximeter.h"
#include  "UbidotsEsp32Mqtt.h"
#include "Adafruit_MLX90614.h"

const char *UBIDOTS_TOKEN = "BBFF-LvNOJ4z5bmM2UESky89WDoqT5OmLCr";
const char *DEVICE_LABEL = "esp32-telemedicina";
const char *OXIGENO_LABEL = "oxigeno";
const char *BMP_LABEL = "heartrate";
const char *TEMPERATURA = "temperatura";

const long PUBLISH_FREQUENCY = 10000; // Update rate in milliseconds

const char *WIFI_SSID = "Telemedicina";      // Put here your Wi-Fi SSID
const char *WIFI_PASS = "Telemedicina.2021*";       // Put here your Wi-Fi password
```

```
long Ts_Last_Publish = 0;

float temperatura = 0.00;
float bpm = 0.0;
float spo2 = 0.0;

Ubidots ubidots(UBIDOTS_TOKEN);
Adafruit_MLX90614 mlx = Adafruit_MLX90614();

PulseOximeter pox;

 void callback(char *topic, byte *payload, unsigned int length){
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++){
    Serial.print((char)payload[i]);
  }
  Serial.println();
}

void onBeatDetected(){
  // Serial.println("Beat!");
}

int scan_devices_i2c(){
  byte error, address;
  int nDevices;
```

```
  Serial.println("Scanning...");
  nDevices = 0;
  for(address = 1; address < 127; address++ ) {
    Wire.beginTransmission(address);
    error = Wire.endTransmission();
    if (error == 0) {
      // Serial.print("I2C device found at address 0x");
      // if (address<16) {
      //  Serial.print("0");
      // }
      Serial.println(address,HEX);
      nDevices++;
    }
    else if (error==4) {
      // Serial.print("Unknow error at address 0x");
      if (address<16) {
        // Serial.print("0");
      }
      Serial.println(address,HEX);
    }
  }
  return nDevices;

}

void setup(){
  Serial.begin(115200);
  delay(200);
```

```
  ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
  ubidots.setCallback(callback);
  ubidots.setup();
  ubidots.reconnect();

  Serial.println('\n');
  Serial.println("Connection established!");
  Serial.print("IP address:\t");
  Serial.println(WiFi.localIP());

  if (!pox.begin()){
    Serial.println("FAILED");
  }else{
    Serial.println("SUCCESS");
  }
  pox.setOnBeatDetectedCallback(onBeatDetected);
  mlx.begin();
}

void loop(){

  pox.update();

  if (millis() - Ts_Last_Publish > PUBLISH_FREQUENCY) {
    if(scan_devices_i2c() > 1){
      Serial.print("Turn-on Temp Gun");
      temperatura = mlx.readAmbientTempC();
```

```
      }else{
        Serial.print("Turn-off Temp Gun");
        temperatura = 0.00;
      }

      bpm = pox.getHeartRate();
      spo2 = pox.getSpO2();
      Serial.print("BPM: ");
      Serial.print(bpm);
      Serial.print("\t");
      Serial.print("SpO2: ");
      Serial.print(spo2);
      Serial.print("\t");
      Serial.print("Temperatura Corporal: ");
      Serial.println(temperatura);
      ubidots.add(BMP_LABEL, bpm);
      ubidots.add(OXIGENO_LABEL, spo2);
      ubidots.add(TEMPERATURA, temperatura);
      ubidots.publish(DEVICE_LABEL);
      Ts_Last_Publish = millis();
    }
  }
```

6) Una vez cargado el código se realizan pruebas con el termómetro digital.

Figura 1: Termómetro digital

7) Se puede visualizar los datos en el monitor del IDE de Arduino.

8) Se observa en Ubidots los datos obtenidos por el termómetro digital.

RESULTADO(S) OBTENIDO(S):

El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y descripciones de cada imagen

CONCLUSIONES:

El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

RÚBRICA DE REVISIÓN DE PRÁCTICA:

Fecha de realización de la práctica:	
Integrantes:	

	Malo	**Regular**	**Bueno**	**Excelente**	**Observaciones**
Sustentación correcta de las prácticas 35% del					

puntaje					
Eficacia, evidencia ilustraciones y correcta organización del progreso de la práctica 35% del puntaje					
Des enlaces de las prácticas 30% del puntaje					

PUNTAJE:	**/10**

<table>
<tr><td colspan="2">CARRERA: INGENIERÍA ELECTRÓNICA</td><td>ASIGNATURA:</td></tr>
<tr><td>NRO. PRÁCTICA:</td><td></td><td>TÍTULO PRÁCTICA: Configuración de servicios web en la nube con Amazon Web Service de prototipo de banco de pruebas de Telemedicina e implementación de cámara web con ESP32.</td></tr>
<tr><td colspan="3">OBJETIVO GENERAL:

Aprender la cconfiguración de servicios web en la nube con Amazon Web Service de prototipo de banco de pruebas de Telemedicina e implementación de cámara web con ESP32 para monitoreo visual remoto.

OBJETIVOS ESPECÍFICOS:

• Conectorizar cámara ESP32 para visualización de pacientes que usarán el prototipo de telemedicina.
• Almacenar en Amazon Bucket S3 los datos obtenidos de los EKG y demás sensores.</td></tr>
<tr><td colspan="2" rowspan="3">INSTRUCCIONES</td><td>1. Los estudiantes deben leer previamente el manual de práctica para el desarrollo.</td></tr>
<tr><td>2. Los estudiantes deben utilizar los equipos del banco de pruebas de telemedicina de una manera responsable y calificada para evitar daños los equipos.</td></tr>
<tr><td>3. Los estudiantes deben trabajar en</td></tr>
</table>

	grupo para el desarrollo de la práctica.
	4. Se debe dejar en orden el sitio de práctica luego del desarrollo de esta.

ACTIVIDADES POR DESARROLLAR:

En esta práctica se explica la configuración de los servicios de Amazon Web Service específicamente con el servicio de almacenamiento Amazon Bucket S3.

EKG_Olimex – Ubidots – Amazon Bucket S3

1) Para la integración de Ubidots con Amazon S3 se debe descargar el plugin de Amazon S3 dentro de la plataforma de Ubidots.

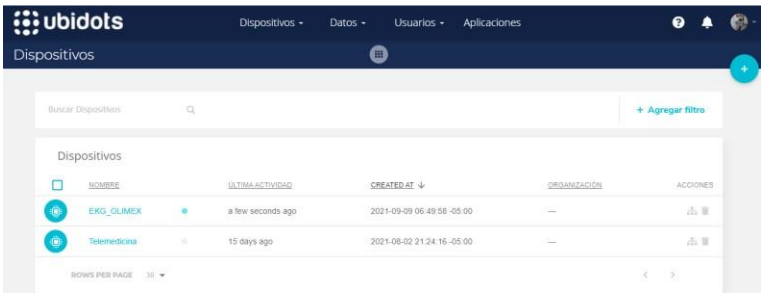


Figura 1: Udidots

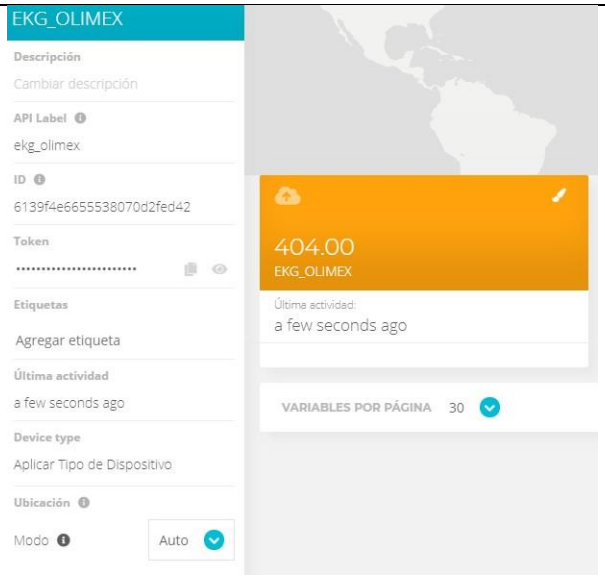


Figura 2: Datos de EKG en Ubidots

FECHA	VALOR	CONTEXT
		New data available
2021-09-14 23:47:17 -05:00	675.00	{}
2021-09-14 23:47:17 -05:00	570.00	{}
2021-09-14 23:47:17 -05:00	596.00	{}
2021-09-14 23:47:17 -05:00	644.00	{}
2021-09-14 23:47:17 -05:00	480.00	{}
2021-09-14 23:47:17 -05:00	449.00	{}
2021-09-14 23:47:17 -05:00	254.00	{}
2021-09-14 23:47:17 -05:00	306.00	{}
2021-09-14 23:47:17 -05:00	341.00	{}
2021-09-14 23:47:17 -05:00	487.00	{}

ROWS PER PAGE 10

Figura 3: Datos almacenados del EKG en Ubidots.

2) Se observa la gráfica del EKG en el dashboard EKG Olimex de Ubidots.

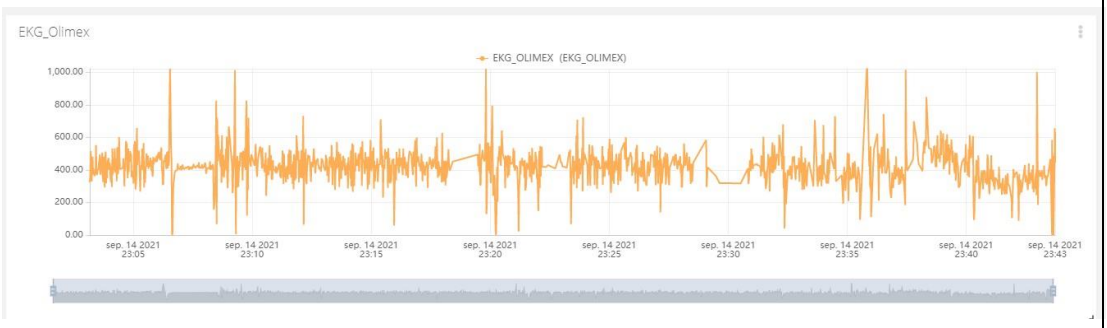


Figura 4: EKG Olimex

3) Se puede observar que las gráficas del EKG directamente desde el monitor plotter del IDE de Arduino es más estable y de mejor resultado.

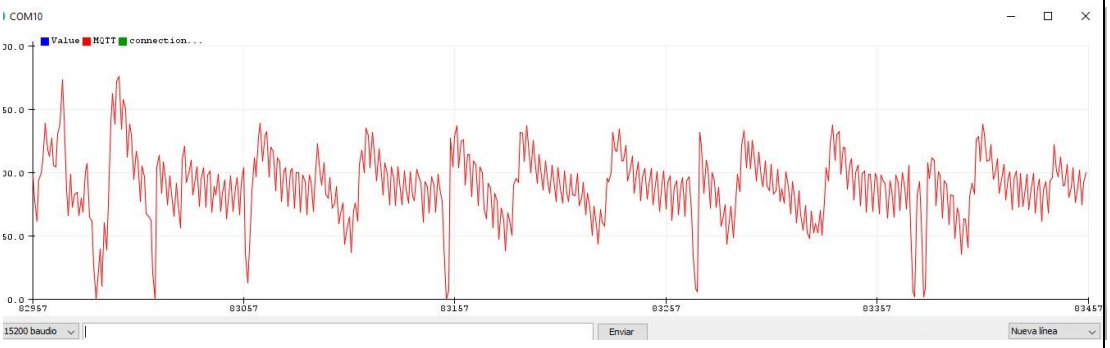


Figura 5: Monitor Plotter EKG

Código EKG-Olimex

```
//UNIVERSIDAD POLITÉCNICA SALESIANA - SEDE GUAYAQUIL
//INGENIERÍA EN ELECTRÓNICA
//PROTOTIPO DE TELEMEDICINA - IOT
//AUTORES: MIGUEL ORTIZ - CRISTIAN MACIAS

// EKG - UDIDOTS

//LIBRERIAS
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiUdp.h>
#include <PubSubClient.h>
#include <NTPClient.h>

//DEFINIR VARIABLES
#define WIFISSID "Telemedicina" // Enter WifiSSID here
#define PASSWORD "Telemedicina.2021*" // Enter password here
#define TOKEN "BBFF-LvNOJ4z5bmM2UESky89WDoqT5OmLCr" // Ubidots' TOKEN

#define MQTT_CLIENT_NAME "mymqttclient" // MQTT client Name
// * Define Constants
#define VARIABLE_LABEL "ekg_olimex" // ubidots variable label
#define DEVICE_LABEL "ekg_olimex" // ubidots device label

const int analogInPin = 39; // Output Olimex is connected to GPIO35
```

```
char mqttBroker[]  = "industrial.api.ubidots.com";
char payload[10000];
char topic[150];

// Space to store values to send
char str_sensor[10];
char str_millis[20];
double epochseconds = 0;
double epochmilliseconds = 0;
double current_millis = 0;
double current_millis_at_sensordata = 0;
double timestampp = 0;
int j = 0;

WiFiClient ubidots;
PubSubClient client(ubidots);
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");

/****************************************
  Auxiliar Functions
 ****************************************/
void callback(char* topic, byte* payload, unsigned int length) {
  char p[length + 1];
  memcpy(p, payload, length);
  p[length] = NULL;
  Serial.write(payload, length);
  Serial.println(topic);
```

```
}
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.println("Attempting MQTT connection...");
    // Attemp to connect
    if (client.connect(MQTT_CLIENT_NAME, TOKEN, "")) {
      Serial.println("Connected");
    } else  {
      Serial.print("Failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 2 seconds");
      // Wait 2 seconds before retrying
      delay(2000);
    }
  }
}

/**************************************
   Main Functions
 **************************************/
void setup() {
  Serial.begin(115200);
  analogReadResolution(10);
  WiFi.begin(WIFISSID, PASSWORD);
  Serial.println();
  Serial.print("Waiting for WiFi...");
  while (WiFi.status() != WL_CONNECTED) {
```

```
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.println("WiFi Connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  timeClient.begin();
  client.setServer(mqttBroker, 1883);
  client.setCallback(callback);
  timeClient.update();
  epochseconds = timeClient.getEpochTime();
  epochmilliseconds = epochseconds * 1000;
  Serial.print("epochmilliseconds=");
  Serial.println(epochmilliseconds);
  current_millis = millis();
  Serial.print("current_millis=");
  Serial.println(current_millis);
}

void loop() {
  if (!client.connected()) {
    reconnect();
    j = 0;
  }

  j = j + 1;
  // Serial.print("j=");
```

```
            // Serial.println(j);
            sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
            sprintf(payload, "%s", ""); // Cleans the payload
            sprintf(payload, "{\"%s\": [", VARIABLE_LABEL); // Adds the variable label
            float sensor = analogRead(analogInPin);
            dtostrf(float(sensor), 4, 2, str_sensor);
            Serial.print("Value: ");
            Serial.println(sensor);
            current_millis_at_sensordata = millis();
            timestampp = epochmilliseconds + (current_millis_at_sensordata -
current_millis);
            dtostrf(timestampp, 10, 0, str_millis);
            sprintf(payload, "%s{\"value\":%s, \"timestamp\": %s}]}", payload,
str_sensor, str_millis);
            // Serial.println("Publishing data to Ubidots Cloud");
            // Serial.println(payload);
            client.publish(topic, payload);
            delay(10);
          }
```

Configuración en AWS

4) Se realiza la configuración en la consola de AWS, previamente se debe haber configurado la cuenta con el usuario y password de AWS, para obtener acceso a los servicios de Amazon Web Services.

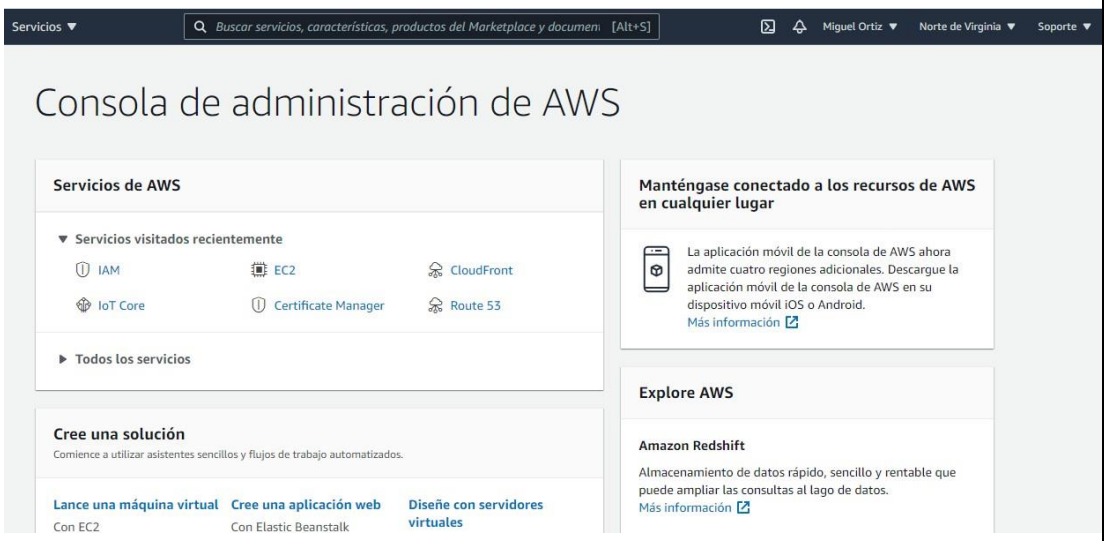


Figura 6: Consola de Administración de AWS

5) Se debe realiza la configuración de accesos y privilegios en IAM de AWS.

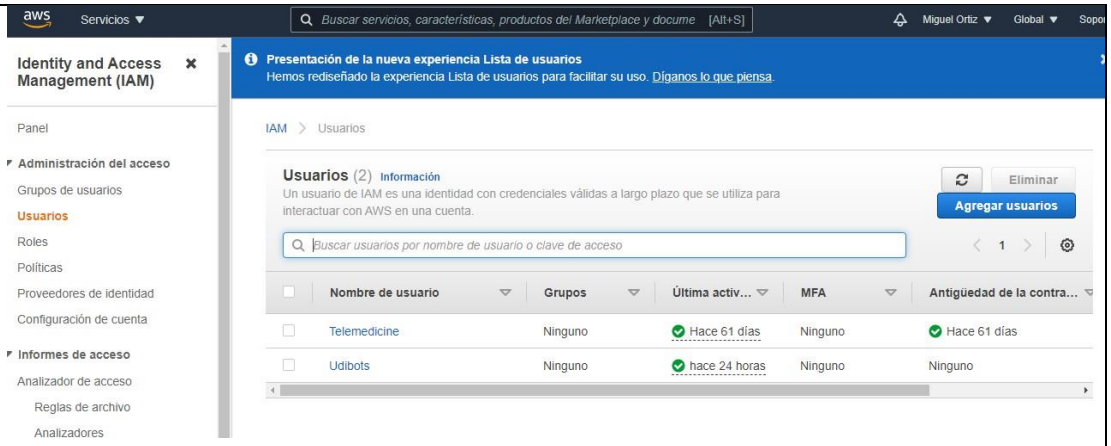


Figura 7: Acceso IAM

6) Se realiza la creación de un bucket S3 en AWS.

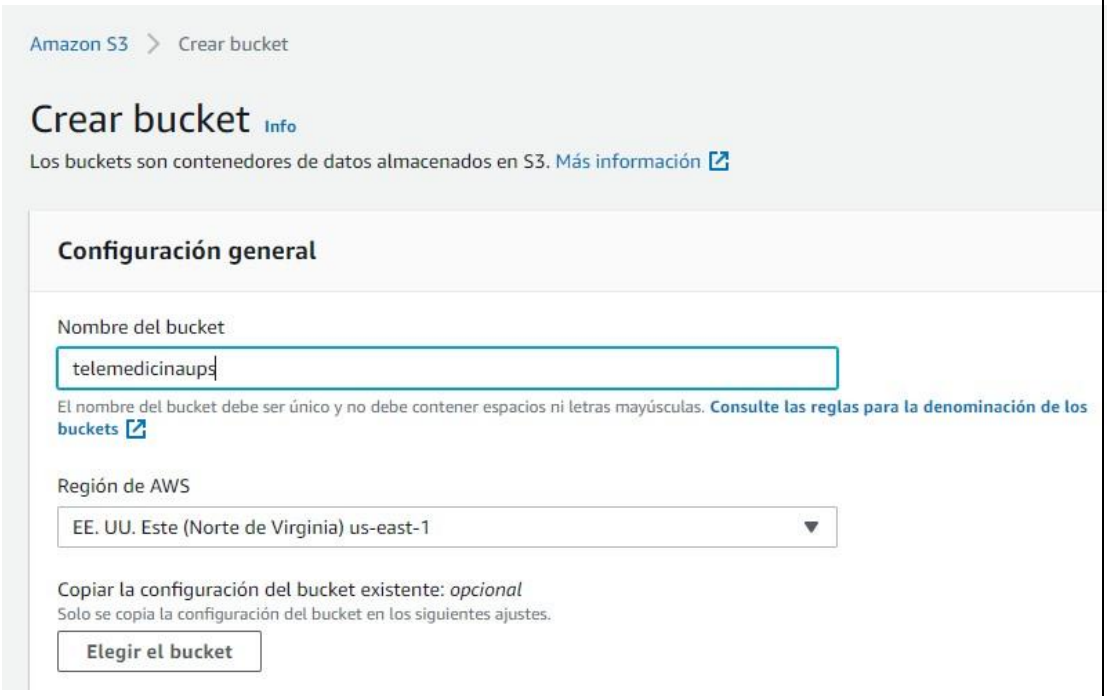


Figura 8: Creación de bucket S3.

7) Se realizan las configuraciones básicas de un bucket S3.

Figura 9: Configuración de bloqueo de acceso del bucket

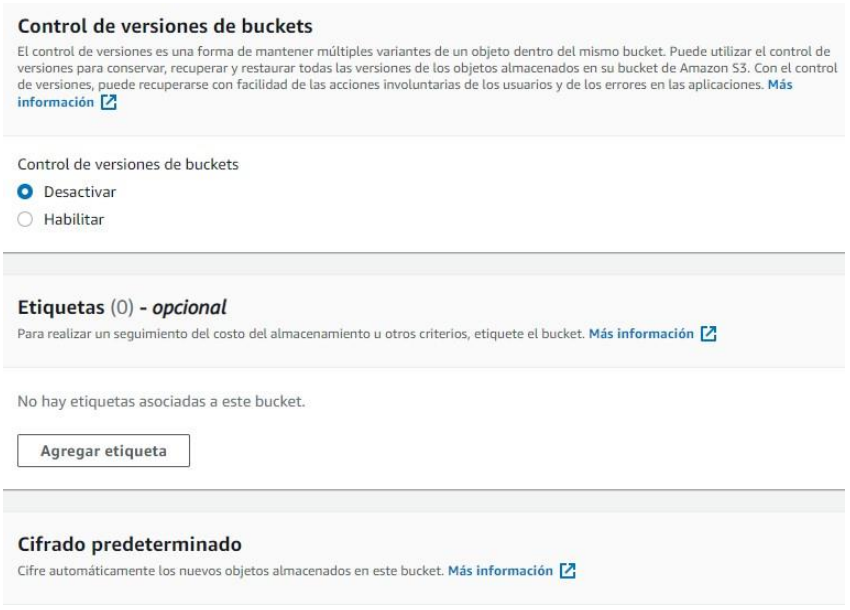


Figura 10: Control de versiones del bucket

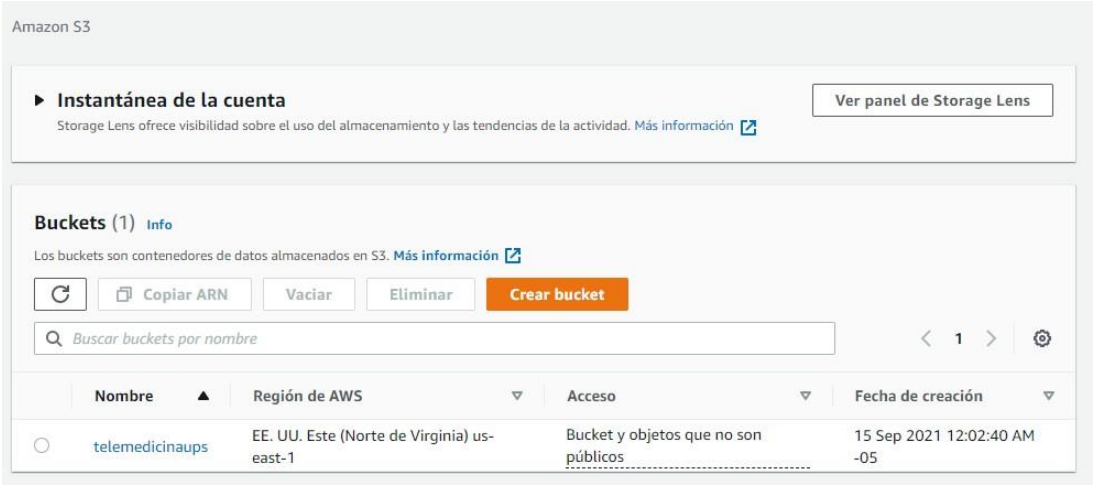


Figura 11: Revisión del bucket creado

8) En Udidots se instala el plugin de AWS S3 Backup.

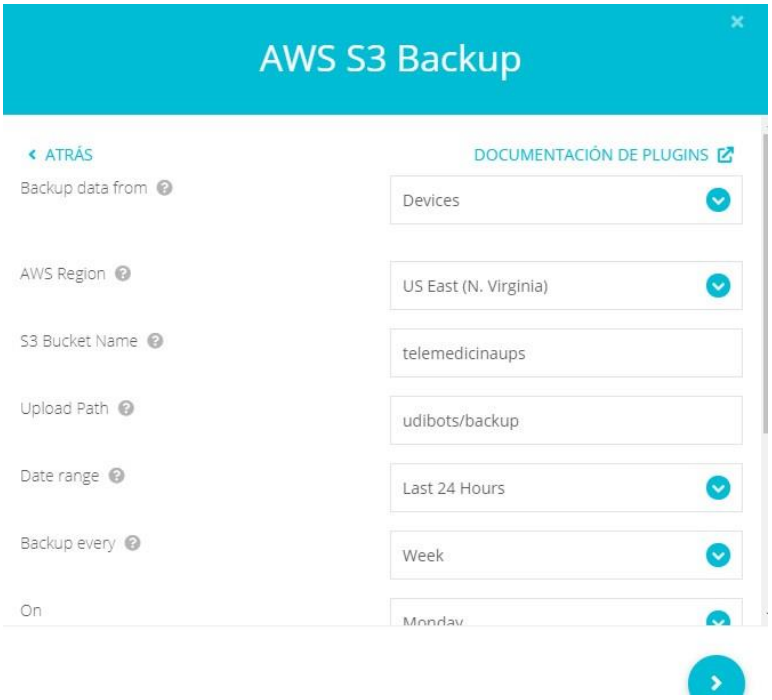


Figura 12: Plugin AWS S3 en Udidots

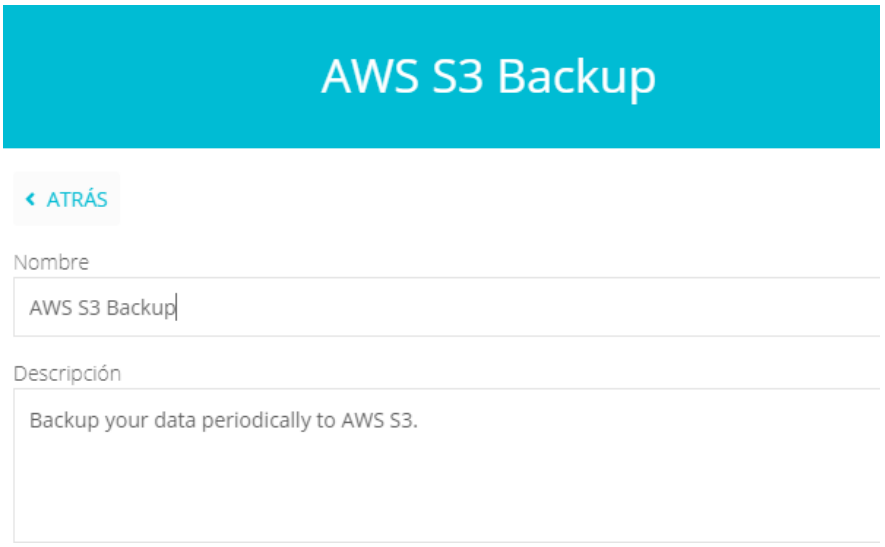


Figura 13: Nombre de AWS S3 Backup

9) Finalmente se valida que el plugin esté correctamente corriendo en la plataforma de Udidots.

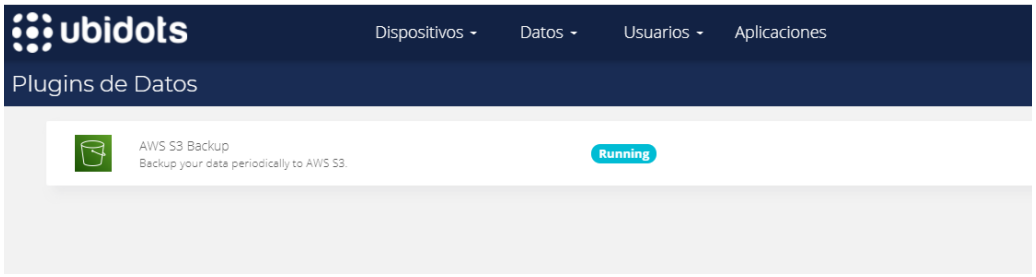


Figura 13: Validación de running de la app

Configuración de cámara web ESP32

10) Para la configuración de la cámara web ESP32Cam se debe cargar el siguiente código en el ESP32Cam.

Código ESP32Cam

```
// UNIVERSIDAD POLITÉCNICA SALESIANA - SEDE GUAYAQUIL
// PROTOTIPO IOT DE TELEMEDICINA
// MIGUEL ORTIZ - CRISTIAN MACIAS
// LAB6 - ESP32-CAM

//LIBRERIAS
#include "esp_camera.h"
#include <WiFi.h>

// Select camera model
//#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

const char* ssid = "Telemedicina";
const char* password = "Telemedicina.2021*";
```

```
void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
```

```
  config.pixel_format = PIXFORMAT_JPEG;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  //                     for larger pre-allocated frame buffer.
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
```

```
    if (s->id.PID == OV3660_PID) {
      s->set_vflip(s, 1); // flip it back
      s->set_brightness(s, 1); // up the brightness just a bit
      s->set_saturation(s, -2); // lower the saturation
    }
    // drop down frame size for higher initial frame rate
    s->set_framesize(s, FRAMESIZE_QVGA);

  #if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
    s->set_vflip(s, 1);
    s->set_hmirror(s, 1);
  #endif

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    startCameraServer();

    Serial.print("Camera Ready! Use 'http://");
    Serial.print(WiFi.localIP());
    Serial.println("' to connect");
```

```
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}
```

11) Se debe validar previamente que la ESP32Cam se reconozca por usb en la PC donde se cargará el código.

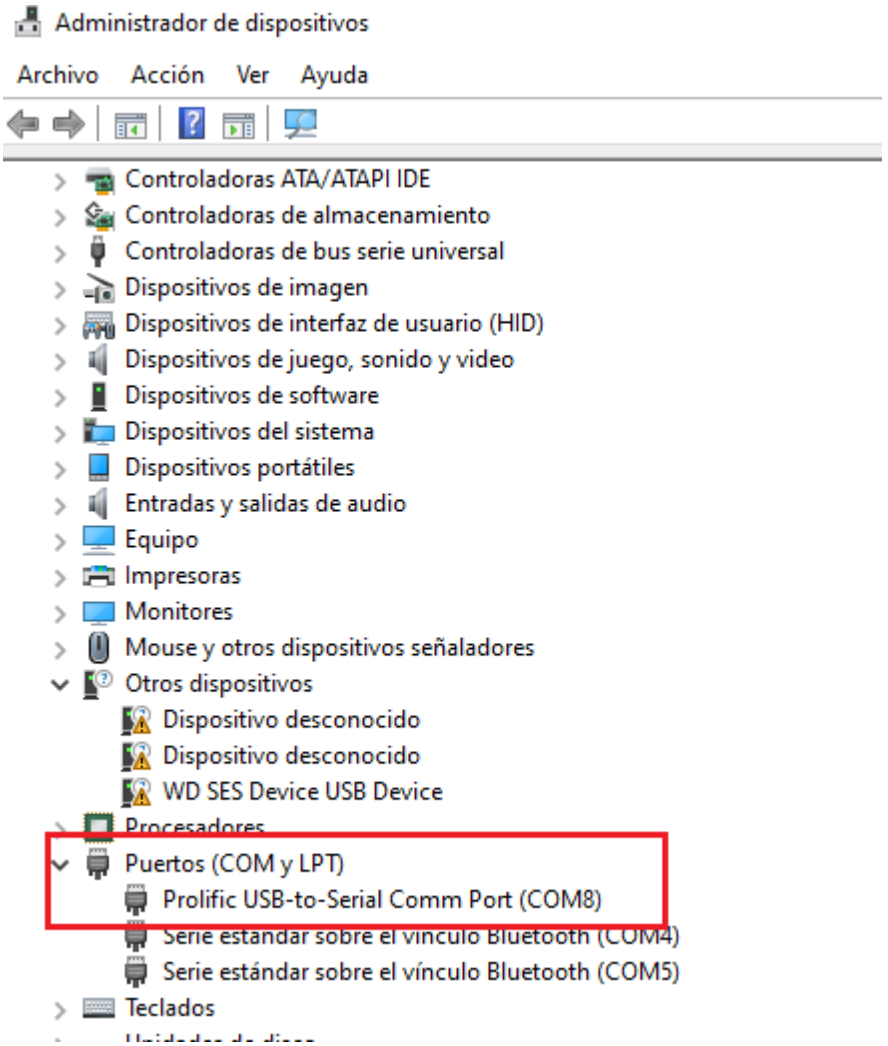


Figura 14: Validación de conexión USB con el ESP32Cam

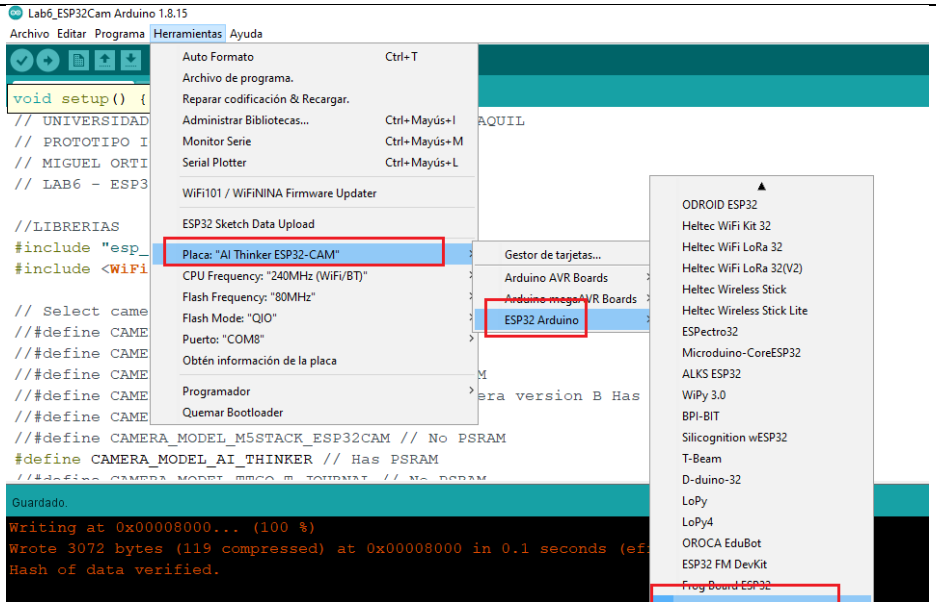


Figura 15: Selección de ESP32 Arduino

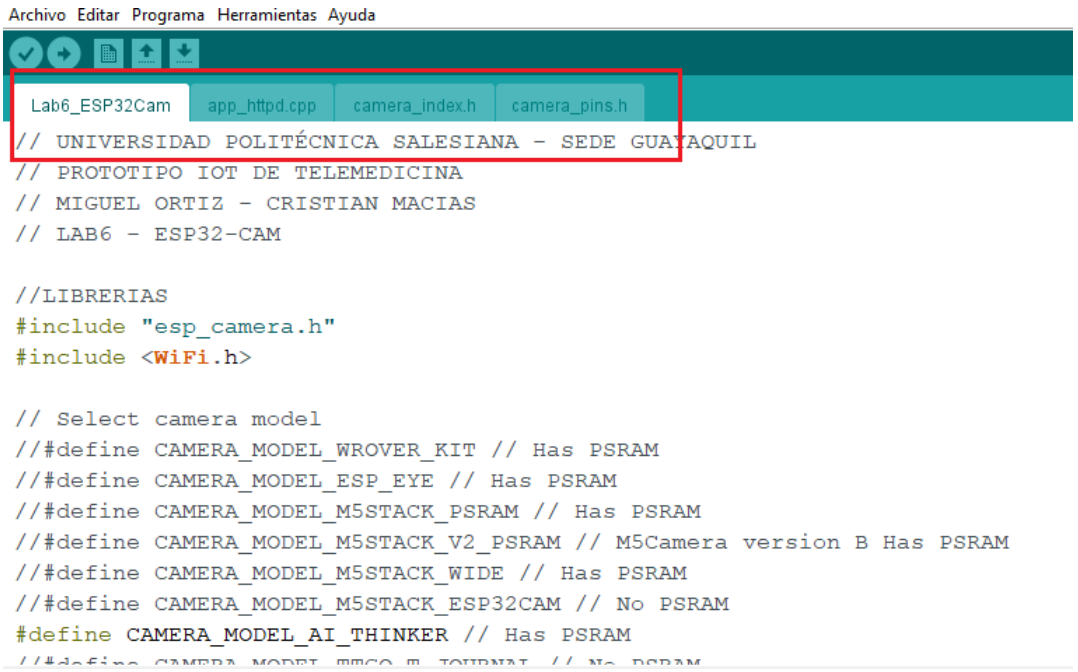


Figura 16: Librerías de ESP32Cam

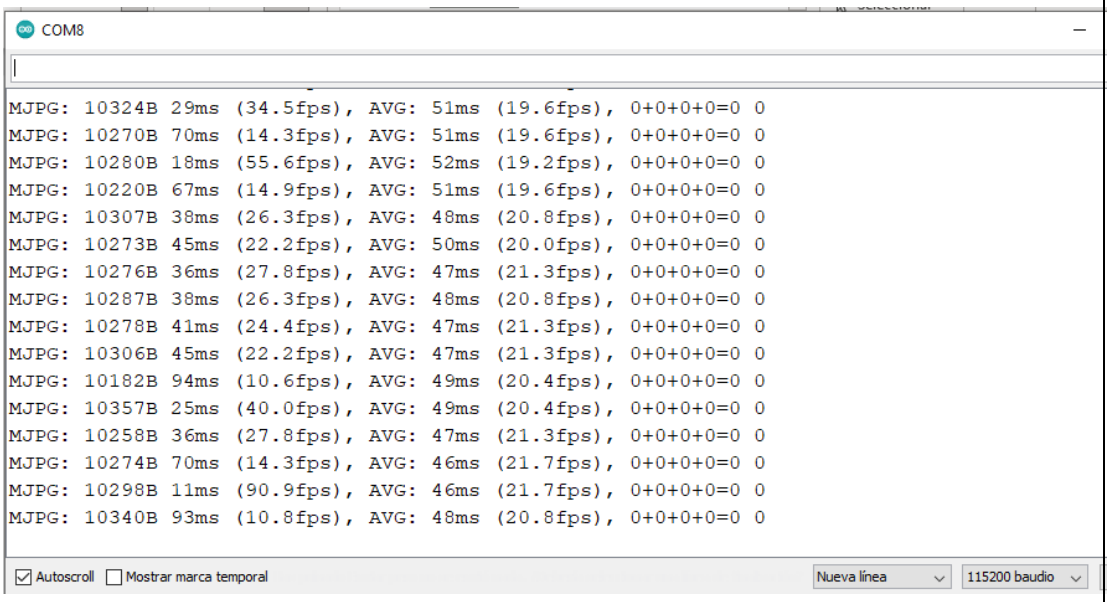


Figura 17: Datos del ESP32Cam en monitor Arduino

12) Luego de la conexión se debe validar que se llega a la ip del ESP32Cam.

13) Abrir vía web con la ip asignada a la cámara ESP32.

14) Configurar los parámetros de visualización de la cámara.

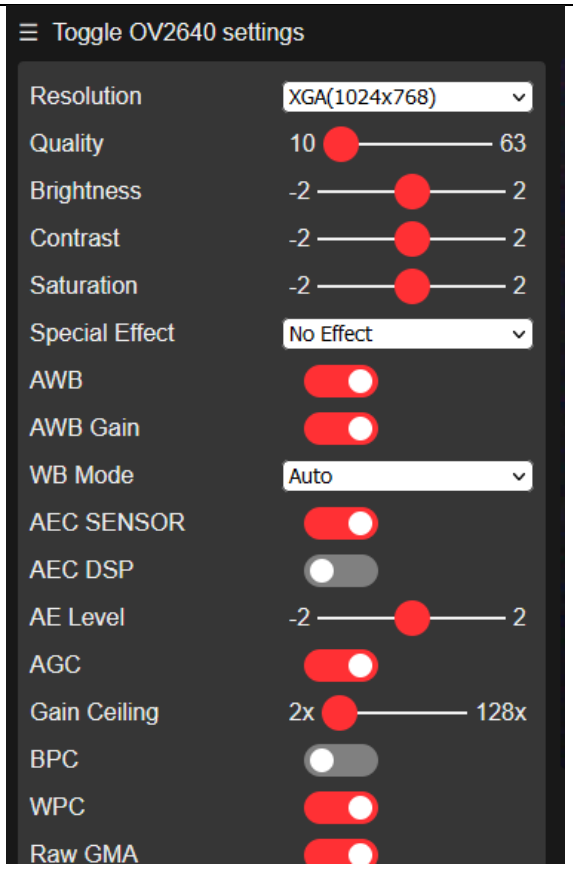


Figura 17: Configuraciones del ESP32Cam.

15) Para poder visualizar externamente la cámara se debe realiza una redirección de puertos en el router donde se conecta a la red la ESP32Cam.

16) Se utiliza un router Mikrotik hap Lite.

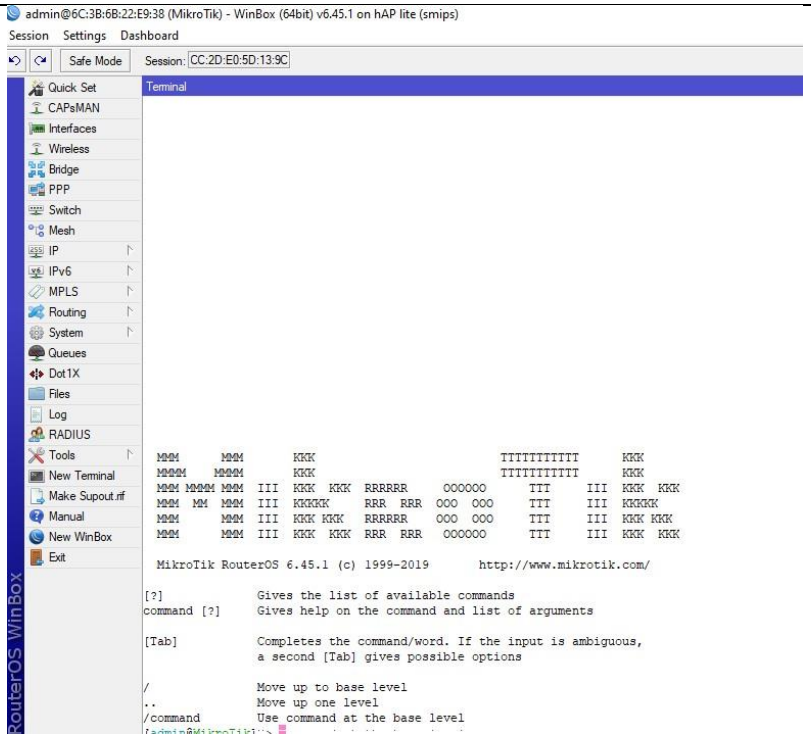


Figura 18: Configuraciones en el router Mikrotik

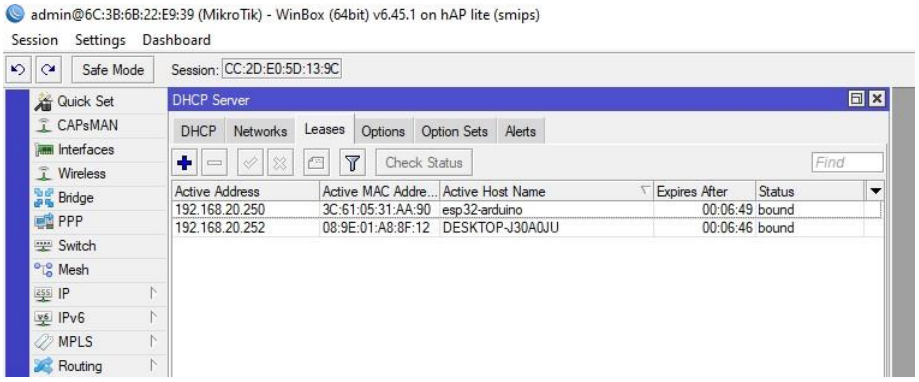


Figura 19: Configuraciones de redireccionamiento de puerto.

RESULTADO(S) OBTENIDO(S):

El estudiante debe colocar los resultados obtenidos en la práctica con imágenes y descripciones de cada imagen

CONCLUSIONES:

El estudiante debe colocar las conclusiones de las prácticas de acuerdo con los objetivos planteados.

RÚBRICA DE REVISIÓN DE PRÁCTICA:

Fecha de realización de la práctica:	
Integrantes:	

	Malo	**Regular**	**Bueno**	**Excelente**	**Observaciones**
Sustentación correcta de las prácticas 35% del puntaje					
Eficacia, evidencia ilustraciones y correcta organización del					

progreso de la práctica 35% del puntaje						
Des enlaces de las prácticas 30% del puntaje						
					PUNTAJE:	**/10**