UCUENCA

Facultad de Ingeniería

Carrera de Electrónica y Telecomunicaciones

Diseño e implementación de un sistema de monitorización de cambios posturales empleando sensores inerciales para la prevención de úlceras por presión

Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones

Autores:

Christian Simón Cullquicondo Gahona

C.I: 0107061400

simon12@live.com

Byron Paúl Sanchez Guerrero

C.I: 0106094410

byronpsanchez4@gmail.com

Director:

Dr. Luis Ismael Minchala Ávila

C.I: 0301453486

Cuenca, Ecuador

26/09/2022

Resumen

Hoy en día existe gran carga laboral en los sistemas de salud, esto conlleva a que el personal no esté demasiado pendiente de los pacientes y sus necesidades. Tal es el caso de pacientes encamados con movilidad limitada que requieren un posicionamiento distinto durante el día. No hacer esto, promueve la aparición de lesiones en la piel conocidas como úlceras por presión, que potencialmente afectan la salud del paciente e incluso su calidad de vida a largo plazo. Con esta consideración, este trabajo propone el diseño e implementación de un sistema que monitorea la posición de pacientes encamados, con la finalidad de alertar al personal médico y prevenir de esta forma la aparición de las úlceras por presión.

El sistema comprende un artefacto ajustable en el pecho del paciente que contiene un dispositivo electrónico capaz de capturar componentes angulares de acuerdo a la posición del paciente. Los datos obtenidos se procesan a través del algoritmo de clasificación k-NN obteniendo información posicional. Esta información es visible desde una aplicación web que permite llevar a cabo un monitoreo del paciente en tiempo real. El sistema es evaluado tanto en personas sanas como pacientes, con resultados alentadores en torno a la clasificación de posturas. Debido a que los sujetos no están propensos a una gran cantidad de movimiento y a que los valores angulares que determinan cada posición están lo suficientemente separados en la gráfica de dispersión, la posición reportada por el dispositivo en todos los casos es igual a la posición real de la persona.

Palabras clave: IMU. IoT. KNN. MQTT. UPP.

Abstract

Today there is a heavy workload in health systems; the staff is not too aware of patients and their needs. Such is the case of bedridden patients with limited mobility who require different positioning during the day. Not doing this promotes the appearance of skin lesions known as pressure ulcers, potentially affecting the patient's health and even their quality of life in the long term. With this in mind, this work proposes the design and implementation of a system that monitors the position of bedridden patients to alert medical personnel and thus prevent the appearance of pressure ulcers.

The system comprises an adjustable artifact on the patient's chest that contains an electronic device capable of capturing angular components according to the patient's position. The data obtained is processed through the k-NN classification algorithm, obtaining positional information. This information is visible from a web application that allows real-time patient monitoring. The system is evaluated in healthy people and patients, with encouraging results regarding posture classification. The classifier's performance is very high according to the statistical analysis of the experimental results. Since people are not prone to a large amount of movement and since the angular values that determine each position are sufficiently separated in the scatterplot, the position reported by the device in all cases is equal to the actual position of the person.

Keywords: IMU. loT. KNN. MQTT. UPP.

Índice general

Índice de figuras

Índice de tablas

UCUENCA

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Christian Simón Cullquicondo Gahona en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Diseño e implementación de un sistema de monitorización de cambios posturales empleando sensores inerciales para la prevención de úlceras por presión", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 26 de septiembre de 2022

Christian Simón Cullquicondo Gahona

C.I: 0107061400

UCUENCA

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Byron Paúl Sánchez Guerrero en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Diseño e implementación de un sistema de monitorización de cambios posturales empleando sensores inerciales para la prevención de úlceras por presión", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 26 de septiembre de 2022

Byron Paúl Sánchez Guerrero

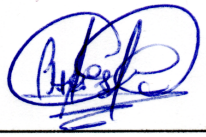
C.I: 0106094410

UCUENCA

Cláusula de Propiedad Intelectual

Christian Simón Cullquicondo Gahona, autor del trabajo de titulación "Diseño e implementación de un sistema de monitorización de cambios posturales empleando sensores inerciales para la prevención de úlceras por presión", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 26 de septiembre de 2022



Christian Simón Cullquicondo Gahona

C.I: 0107061400

UCUENCA

Cláusula de Propiedad Intelectual

Byron Paúl Sánchez Guerrero, autor del trabajo de titulación "Diseño e implementación de un sistema de monitorización de cambios posturales empleando sensores inerciales para la prevención de úlceras por presión", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 26 de septiembre de 2022

Byron Paúl Sánchez Guerrero

C.I: 0106094410

Agradecimientos

Quiero agradecer primeramente a Dios por haberme bendecido con la fortaleza y la sabiduría necesarias para llegar a la meta, luego de un largo y duro camino.

De igual manera agradezco a mis padres y hermanos por darme su apoyo incondicional a lo largo de estos años, ellos son la principal motivación que tengo para salir adelante. Les agradezco por cada momento vivido porque eso me ha servido para llegar hasta aquí.

También quiero agradecer a la Universidad de Cuenca y sus docentes por abrirme sus aulas y guiarme para ser un gran profesional.

Finalmente, agradezco a todos los familiares, amigos y personas especiales en mi vida por creer en mi, darme ánimos y apoyarme de una u otra forma hasta lograr esta meta.

Christian Simón Cullquicondo Gahona

Agradecimientos

Primeramente doy gracias a mis padres quienes me han apoyado a lo largo de esta travesía, siendo un pilar fundamental para cumplir este objetivo que parecía tan lejano en su momento.

De igual manera a mis hermanos, que siempre fueron uno de los motivos principales para conseguir este logro.

Finalmente, a los docentes y a la institución por sus esfuerzos otorgados en mi formación universitaria.

Byron Paúl Sánchez Guerrero

Abreviaciones y Acrónimos

ADC Analog to Digital Converter. 29

BCS Body Coordinate System. 26

DMP Digital Motion Processor. 24

HTML HyperText Markup Language. 46, 47

HTTP Hypertext Transfer Protocol. 34

I2C Inter Integrated Circuits. 23–25, 40

IDE Integrated Development Environment. 30

IMU Inertial Measurement Unit. 23, 26, 39, 40, 42, 49

IoT Internet of Things. 23, 29, 33, 35

JSON JavaScript Object Notation. 47

k-NN k-Nearest Neighbor. 35, 36, 44, 58

LED Light-Emitting Diode. 49

MQTT Message Queuing Telemetry Transport. 33–35, 42, 43

PCB Printed Circuit Board. 49, 50

PHP Hypertext Preprocessor. 46, 47

RDP Remote Desktop Protocol. 30

RGB Red, Green, Blue. 49

SBC Single Board Computer. 30

SCL System Clock. 24

SDA System Data. 24

SMT Surface Mount Technology. 29

SSH Secure Shell. 30

SSL Secure Sockets Layer. 34

SVM Support vector machine. 35

TLS Transport Layer Security. 34

UPP Ulceras por Presión. 18–22, 36–38, 62

USB Universal Serial Bus. 52

Wifi Wireless Fidelity. 42

1. Introducción

Este capítulo presenta los antecedentes, problema y justificación que motivó la elaboración de esta tesis. Además, aborda los objetivos generales, específicos y las principales contribuciones del proyecto.

1.1. Antecedentes

Existen muchas situaciones clínicas en las que la piel y los tejidos blandos soportan gran cantidad de peso de manera continua, tal es el caso de pacientes inmovilizados o con reposo prolongado en cama [1]. La exposición a una carga o fricción sostenida en varios casos ocasiona la ruptura de los tejidos blandos en áreas vulnerables, típicamente sobre prominencias óseas, lo que lleva al desarrollo de Ulceras por Presión (UPP) [2].

Según [3], las UPP son una carga de gran relevancia para los sistemas de salud a nivel mundial y representan entre el 5-12 %, mientras en América es del 7 % de acuerdo a la OMS [4]. En Europa, según datos de estudios de prevalencia, a nivel hospitalario en España se ha estimado entre 7-8 %, Italia (8.3 %), Alemania (10.2 %), Suecia (22.9 %), Francia (8.9 %), Portugal (12.5 %), Reino Unido (21.9 %), Dinamarca (22.7 %). Resulta interesante la baja prevalencia en hospitales de China (1.5 %), a diferencia de EEUU (14-17 %), Chile (28 %), Brasil (39.81 %) y Colombia (26.7 %) [5]. Las UPP son muy incómodas y costosas de tratar, desencadenando consecuencias como amputaciones e inclusive la muerte [6]. Las estadísticas indican que las úlceras por presión causaron 29.000 muertes solo en 2013, con el paso del tiempo esta cantidad aumenta [7].

1.2. Identificación del problema

Un análisis de costos de las UPP muestran que el valor del tratamiento aumenta con la gravedad de la úlcera, debido al mayor tiempo de curación y recuperación de los pacientes. Los costos incluyen gastos en insumos, tiempo de enfermería y cargos de hospitalización, generando así un impacto importante al sistema de salud [8]; ante esto, la prevención surge como una medida muy útil para atacar este problema. Según [9], los cambios de postura son el método principal para prevenir las úlceras por presión. Por lo tanto, monitorear los cambios de postura a lo largo del tiempo y alertar a los cuidadores en caso de requerirse un reposicionamiento resulta una herramienta fundamental en la prevención de las UPP.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar e implementar un sistema de monitorización en tiempo real de cambios posturales en pacientes encamados, utilizando sensores inerciales e inteligencia artificial a fin de prevenir la aparición de UPP.

1.3.2. Objetivos específicos

- Construir un prototipo a través de la integración de unidades de medición inercial y una plataforma de desarrollo que identifique cambios de posición a través de un algoritmo de clasificación.
- Diseñar una aplicación web en conjunto con una base de datos para el monitoreo, alerta de cambios posturales y gestión de información de los pacientes.
- Realizar pruebas de funcionamiento del sistema en un entorno controlado a sujetos sanos y pacientes ideando distintos escenarios riesgosos.
- Analizar los resultados de operación del sistema implementado en torno a la clasificación de posturas y alertas establecidas.

1.4. Contribución

A continuación se detallan las principales contribuciones realizadas en este trabajo de titulación.

- Diseño e implementación de un dispositivo tecnológico que permite monitorizar cambios posturales de pacientes encamados, a través de sensores inerciales y un algoritmo clasificador para prevenir la aparición de UPP.
- Automatización del test de Braden para determinar de forma objetiva el riesgo de aparición de UPP.

2. Estado del arte

Este capítulo describe trabajos relacionados con diversas metodologías tecnológicas enfocadas en la prevención de las UPP, con énfasis en las estrategias utilizadas por los autores, además de sus resultados.

2.1. Trabajos relacionados

El desarrollo tecnológico en conjunto con las iniciativas de investigación para la prevención de enfermedades, conllevan al diseño de sistemas inteligentes que permiten mejorar la calidad de vida de las personas y reducir costos de atención médica.

En torno a la prevención de UPP existen trabajos con soluciones simples de hace algunos años como [10], donde se presenta el diseño de una cama que comprende un eje longitudinal coincidiendo con la línea media imaginaria del colchón y unido a la cama. Ésta se mueve de derecha a izquierda y viceversa con la ayuda de un motor eléctrico que está unido al eje para alternar los puntos de presión del paciente. Por otro lado, [11] implementa un asiento conformado por 6 líneas hidráulicas con acción electromecánica que produce movimiento peristáltico. Este produce redistribución de la presión y estimulación por movimiento sobre la superficie de apoyo.

A día de hoy, existen algunos trabajos de base tecnológica más innovadores que tienen como objetivo mejorar el cuidado y monitorear cambios posturales para la prevención de UPP. Tal es el caso de [12] con el diseño de un dispositivo que establece y notifica la frecuencia de movimiento para prevenir UPP y caídas. Este sistema usa sensores inerciales para la medición de ángulos que son comparados para identificar la postura del paciente. Relacionado a esto, en [13] se plantea una frecuencia de cambio de posición cada 2 horas en colchones estándar, mientras que cada 3-4 horas en superficies especiales viscoelásticas en población de riesgo de UPP. El trabajo presentado en [14] emplea técnicas de inteligencia artificial para el monitoreo de cambios posturales usando sensores inerciales, además integra un modelo basado en datos para clasificar las posturas en la cama a partir de sensores que se encuentran en las prendas del paciente. Un modelo difuso basado en conocimiento calcula la prioridad de los cambios posturales para las zonas del cuerpo y obtiene un rendimiento positivo en la clasificación de las posturas en la cama y una alta adaptabilidad del enfoque difuso.

De manera similar [15] presenta un sistema inteligente para la prevención de UPP, mediante el monitoreo de cambios posturales con dos sensores inerciales portátiles. Los

dispositivos en conjunto permiten estimar la orientación del cuerpo de un paciente a través de algoritmos de aprendizaje automático. El sistema lleva un registro histórico de la actividad corporal e incorpora un envío de alertas cuando un paciente persiste en una posición por un periodo prolongado de tiempo. En la misma línea de los sistemas inteligentes, [16] propone un diseño e implementación no invasivo de sensores portátiles para la prevención de UPP a través de técnicas de aprendizaje profundas. Además, utiliza sensores inerciales con los que estima las posiciones de los pacientes y de igual manera envía un mensaje de alerta.

En ciertas ocasiones existe más de un dispositivo, tal es el caso de [17]. Este proyecto considera dos prototipos. Por un lado, se tiene el PUMP1, que este es un dispositivo electrónico portátil adherido a la bata del paciente sin contacto con la piel. El segundo es el denominado PUMP2 que consiste en cuatro dispositivos electrónicos idénticos colocados debajo de las ruedas de la cama del paciente. El estudio clínico demostró la captura exitosa del movimiento de reposicionamiento del paciente por parte de los dispositivos PUMP1 y PUMP2 con un 85 % de confiabilidad, 2 falsos positivos y 11 movimientos perdidos.

Otra alternativa presenta [18], que emplea sensores sensibles a presión colocados en una cama para revelar los puntos de contacto que llevan al desarrollo de UPP. Este sistema incorpora un algoritmo de aprendizaje profundo para la clasificación de las posturas, sus resultados indican una precisión de hasta el 93 %. De manera similar, [19] usa la tecnología de mapeo de presión en conjunto con un algoritmo de aprendizaje profundo para el procesamiento de los datos, el estudio proporcionó una clasificación de las posturas estáticas con una precisión que oscilaba entre el 70 y el 84 %.

Por otra parte, el trabajo desarrollado por [20] tiene un enfoque que identifica las posturas automáticamente usando datos recopilados de un sistema de mapeo de presión disponible comercialmente, donde usa redes neuronales profundas para la clasificación de las posturas. Este trabajo logró obtener una precisión de hasta el 98 % al clasificar cinco posturas diferentes en la cama para más de 60.000 imágenes de presión. En cuanto a [21], este propone una solución híbrida entre un sistema para la detección y cuantificación de la posición usando tiras con sensores sensibles a la fuerza y el uso de una cámara inteligente. Las tiras están colocadas bajo el paciente en zonas de presión específicas y la cámara emplea un algoritmo de procesamiento de imágenes que mejora la precisión al detectar movimientos del paciente.

Desde una perspectiva ligeramente diferente a las anteriores mencionadas, [22] aborda el problema con un diseño no centrado en la identificación de la posición del paciente. Por el contrario, desarrolla un sistema de sensor inalámbrico autónomo tipo parche que alerta al personal médico sobre un nivel potencialmente dañino de presión de contacto que sugiera la posibilidad de una UPP.

2.2. Análisis

De acuerdo a los trabajos revisados las UPP constituyen una gran carga para los sistemas de salud a nivel mundial, por lo que encontrar un tipo de solución es algo importante y necesita constante investigación y desarrollo. Existe una cantidad impresionante de soluciones propuestas para tratar el tema, cada una de ellas con distintas tecnologías y maneras de abordar el problema. Con lo analizado, se opta por la tecnología de sensores inerciales como punto de partida para el desarrollo de la solución, esto debido a la facilidad de acceso a dichos dispositivos y sobretodo a la mayor cantidad de información relacionada con proyectos similares.

3. Fundamentos teóricos

Este capítulo contiene una descripción basada en fundamentos teóricos acerca de las características que distinguen al proyecto.

A medida que la presencia del Internet crece en la vida de las personas, el desarrollo de productos inteligentes, capaces de conectarse a la web e intercambiar información entre usuarios u otros dispositivos se vuelve cada vez más común y necesario [23]. El Internet of Things (IoT), se puede explorar en varias áreas; abarca industrias, departamentos de transporte, salud y seguridad, etc [24]. En el desarrollo de proyectos IoT, es de vital importancia el *hardware*, compuesto por sensores, actuadores y dispositivos de comunicación integrado; una capa media para análisis y almacenamiento de datos y una interfaz accesible, adaptada a diferentes plataformas [23].

3.1. Unidad de medición inercial

Una Inertial Measurement Unit (IMU) es un dispositivo electrónico que está formado por un conjunto de sensores que son capaces de medir parámetros físicos, tales como aceleración, velocidad u orientación [25]. Los sensores que componen la IMU son acelerómetros, giroscopios y magnetómetros que pueden estar alineados en cada uno de los ejes del marco de referencia tridimensional [26].

La IMU adecuada para este proyecto surge de un análisis preliminar entre los modelos más comerciales analizados en trabajos anteriores. La tabla 3.1 indica un resumen de las características principales de los modelos en cuestión. En [27] tras analizar factores cómo costo, sensibilidad de los sensores internos, protocolo de transmisión de datos y facilidad de compra, el dispositivo elegido fue el MPU6050. De igual manera, en el proyecto desarrollado en [28], luego de las respectivas pruebas con cada IMU, tomando los sensores inerciales de un *smartphone* como referencia para la validación de datos, las mejores coincidencias las obtuvieron con la MPU6050. Con estas consideraciones y de acuerdo a la disponibilidad en el mercado local la MPU6050 es el dispositivo escogido para el desarrollo de este proyecto.

3.1.1. Comunicación I2C

Inter Integrated Circuits (I2C), es un tipo de bus diseñado por Philips Semiconductors que se utiliza para conectar circuitos integrados, únicamente requiere de dos

Tabla 3.1: Comparación de distintos modelos de IMU

	MPU6050	**MPU9250**	**LSM9DS1**
Sensores	Acelerometro Giroscopio	Acelerometro Giroscopio Magnetómetro	Acelerometro Giroscopio Magnetómetro
Alimentación	3.3V, 5V	5V	
Grados de Libertad (DOF)	6	9	
Rango Acelerómetro (g)	2,4,6,8		
Resolución acelerómetro (bits)	16		
Rango giroscopio (dps)	250 - 2000		245 - 2000
Resolución giroscopio (bits)	16		
Rango magnetómetro (uT)	No tiene	4800	400, 800, 1200, 1600
Resolución magnetómetro (bits)	-	14	16
Interface de comunicación	I2C	I2C, SPI	
Costo	$6	$10	$12
Observaciones	Disponen de un Digital Motion Processor (DMP) integrado que les permite realizar cálculos internamente de parámetros como ángulos Euler, Quaterniones, etc.		

líneas de señal y un común o masa. Permite el intercambio de información entre muchos dispositivos a una velocidad aceptable, de unos 100 Kbits por segundo, aunque hay casos en los que el reloj llega hasta los 3,4 MHz. La metodología de comunicación de datos del bus I2C es en serie y sincrónica. Una de las señales del bus System Clock (SCL) marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos System Data (SDA) [29]. El I2C es un bus con múltiples maestros, lo que significa que se pueden conectar varios chips al mismo bus y que todos ellos pueden actuar como maestro, sólo con iniciar la transferencia de datos. La figura 3.1 muestra la estructura básica de la arquitectura I2C.

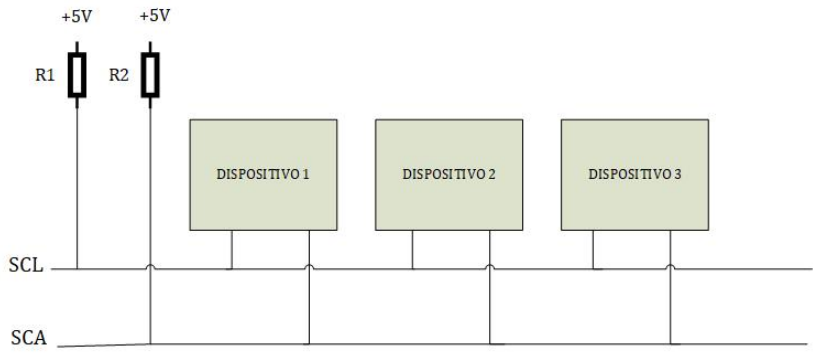


Figura 3.1: Estructura básica de la comunicación I2C

3.1.2. MPU6050

MPU6050 es un sensor que contiene un acelerómetro MEMS y un giroscopio MEMS en un solo chip. Tanto el acelerómetro como el giroscopio contienen 3 ejes que pueden capturar información en X, Y y Z con un ADC de 16 bits para cada canal [30]. Además, tiene una interfaz de bus I2C para comunicarse con microcontroladores. La figura 3.2 muestra la disposición de placa de este dispositivo.

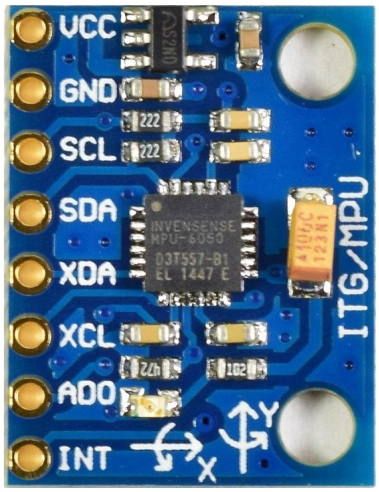


Figura 3.2: Dispositivo MPU6050

MPU6050 consta de 8 pines, cuya función se indica a continuación [31]:

- VCC: Proporciona energía para el módulo, puede ser de +3.3V a +5V.
- GND: Conectado a tierra del sistema.
- SCL: Utilizado para proporcionar pulso de reloj para comunicación I2C.
- SDA: Utilizado para transferir datos a través de la comunicación I2C.
- XDA: Pin de datos en serie auxiliar. Este pin se utiliza para conectar otros sensores habilitados con interfaz I2C.
- XCL: Pin de reloj serie auxiliar. Este pin se utiliza para conectar otros sensores habilitados con interfaz I2C.
- AD0: Este es el bit 0 en la dirección esclava de 7 bits del dispositivo. Si está conectado a VCC, entonces se lee como uno lógico y la dirección del esclavo cambia.
- INT: Pin de salida digital de interrupción.

3.2. Ángulos de inclinación

Los ángulos de inclinación determinan la posición de un objeto en un tiempo determinado a partir de las señales obtenidas de una IMU.

3.2.1. Sistema de coordenadas sólidas

Body Coordinate System (BCS), es un sistema cuyo origen está en el centro de masa del cuerpo analizado. Este sistema es usado con frecuencia en plataformas *strapdown*, es decir, cuando los ejes de los sensores y del cuerpo en donde están montados se mueven en forma conjunta [32]. En el sistema BCS sus 3 ejes fijos son relativos al eje X, Y y Z del plano cartesiano, estableciendo rotaciones intrínsecas principales, es decir, relativas al sistema [33], en la figura 3.3 se aprecia dicha relación:

- Roll: Es una inclinación o rotación sobre el eje X, puede ser en sentido positivo o negativo.
- Pitch: Es una inclinación o rotación alrededor del eje Y, de igual manera puede darse en sentido positivo o negativo.
- Yaw: Rotación respecto al eje vertical Z.

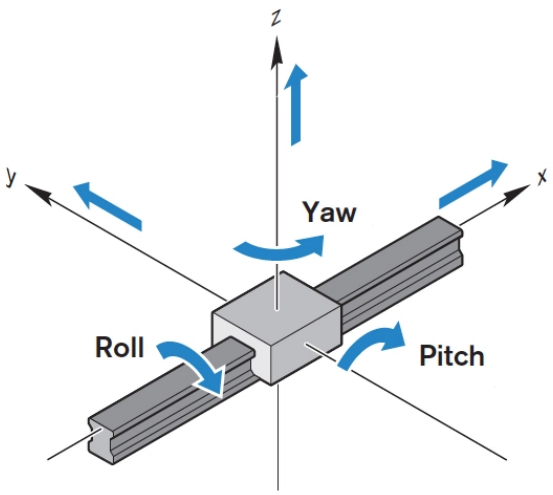


Figura 3.3: Ángulos de rotación en el sistema BCS

3.2.2. Cálculo de inclinación usando giroscopio

El giroscopio mide la velocidad angular de un objeto en cada eje coordenado. Obtiene cada ángulo de giro sobre su propio eje integrando dicha velocidad. La figura 3.4

muestra que la velocidad angular es perpendicular al plano de rotación [34].

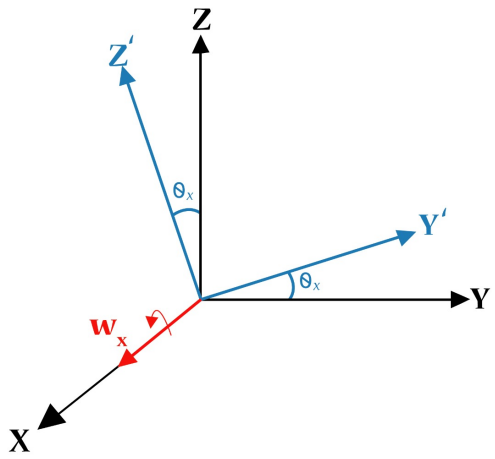


Figura 3.4: Ángulo de rotación sobre el eje X

Matemáticamente esto resulta del producto de la velocidad angular por el tiempo en el que ocurre el movimiento, obteniendo un ángulo instantáneo. Sin embargo, para encontrar el ángulo de inclinación a lo largo del tiempo es necesario establecer intervalos fijos Δt con lo cual, el ángulo es el resultado de un algoritmo acumulativo descrito en la ecuación siguiente para cada eje coordenado:

$$\begin{aligned} \theta_x &= \theta_{x_0} + \omega_x \, \Delta t \\ \theta_y &= \theta_{y_0} + \omega_y \, \Delta t \end{aligned} \tag{3.1}$$

donde, θ_0 corresponde a la posición inicial actualizada en cada iteración.

Existen un inconveniente importante al determinar el ángulo de inclinación únicamente mediante el giroscopio y es el error acumulativo "drift", que aparece a lo largo del tiempo debido a la integración [35].

3.2.3. Cálculo de inclinación usando acelerómetro

Un acelerómetro capta y cuantifica un movimiento de aceleración. El principio de funcionamiento es medir la fuerza de movimiento de una masa por medio de transductores que registran y envían este valor en una señal de voltaje [36]. El acelerómetro funciona como sensor de inclinación, usando la aceleración de la gravedad como un vector para determinar la orientación de un objeto en el espacio. En el caso de 2D, con el acelerómetro ubicado horizontalmente en el plano X-Y, con Z apuntando hacía

arriba y girando en el eje Y, tal y como muestra la figura 3.5, la ecuación para el ángulo resulta:

$$\theta = \tan^{-1}\left(\frac{g_x}{g_z}\right) \tag{3.2}$$

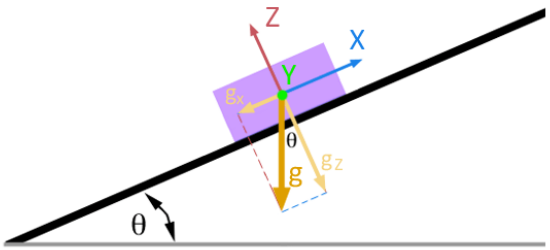


Figura 3.5: Ángulo de inclinación sobre el eje Y medido en 2D

Los valores obtenidos en cada acelerómetro son las componentes de la gravedad que actúan en cada uno de los ejes. Las ecuaciones que consiguen los ángulos de inclinación para un modelo 3D, de acuerdo a lo señalado en [37] son:

$$\begin{aligned} \theta_x &= \tan^{-1}\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \\ \theta_y &= \tan^{-1}\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \end{aligned} \tag{3.3}$$

Los acelerómetros son dispositivos muy sensibles a las vibraciones, por lo que la medición presenta ruido de alta frecuencia debido a movimientos ligeros, esto produce que el ángulo obtenido no sea el acertado todo el tiempo [35].

3.2.4. Cálculo de inclinación con filtro complementario

El filtro complementario es un sistema que reúne las medidas del acelerómetro y giroscopio [38]. Éste aplica un filtro pasa-bajos a la señal del acelerómetro, para eliminar el error producido en los cambios rápidos de posición. Además, aplica un filtro pasa-altos a la señal proveniente del giroscopio que elimina el error acumulativo generado en largos periodos de tiempo, todo esto lo resume el diagrama de la figura 3.6. La ecuación siguiente muestra la fórmula del filtro complementario utilizado para calcular el ángulo requerido, θ, donde α es el factor de fidelidad entre la lectura del giroscopio versus la lectura del acelerómetro ($0 < \alpha < 1$):

$$\theta_{estimado} = \alpha \cdot (\theta_{est\_previo} + \omega_{gyro} \cdot dt) + (1 - \alpha) \cdot \theta_{acel} \tag{3.4}$$

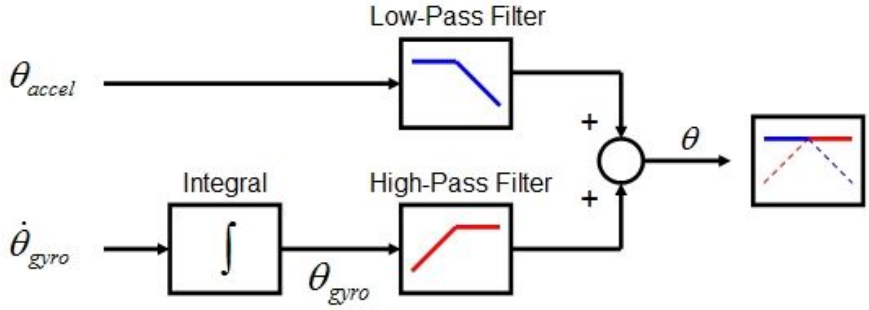


Figura 3.6: Filtro complementario

3.3. Microcontrolador para IoT ESP8266

ESP8266 es el nombre de un microcontrolador diseñado por una compañía china llamada Espressif Systems. De acuerdo a lo señalado por el fabricante en 2014, este chip constituye una excelente solución autómata de redes wifi que se ofrece como puente entre los microcontroladores que hasta ahora existen o que tiene la capacidad de ejecutar aplicaciones independientes [39]. Un ESP8266 salido de fábrica resulta de mucha utilidad, ya que su producción está basada en la compactación de un chip Surface Mount Technology (SMT) el cual viene en un pequeño paquete de tan solo cinco milímetros cuadrados. Este chip está integrado en placas de circuito impreso prefabricadas donde se lo adecua y queda listo para su uso [40]. El módulo ESP8266 es ideal para aplicaciones IoT por su bajo costo, características, variantes y aplicaciones.

El ESP8266 tiene un pin Analog to Digital Converter (ADC) con una resolución de 10 bits que varía la lectura analógica de 0 a 1023 [41]. El ADC solo convierte el voltaje entre 0 y 1V. La arquitectura de ahorro de energía del ESP8266 funciona en 3 modos: modo activo, modo de suspensión, modo de suspensión profunda [42]. La integración de dispositivos específicos de la aplicación o los sensores con la placa ESP8266 es fácil a través de sus pines GPIO que crean una forma de conectar la placa ESP8266 al mundo externo [43].

Wemos D1 Mini ESP8266 es una plataforma de desarrollo similar a Arduino especialmente orientada al IoT (Vea la figura 3.7). La placa Wemos D1 Mini ESP8266 tiene como núcleo al SoM ESP-12E que a su vez está basado en el SoC Wi-Fi ESP8266, integra además el conversor USB-Serial TTL CH340G y conector micro-USB necesario para la programación y comunicación a PC. Wemos D1 mini está diseñado especialmente para trabajar montado en protoboard o soldado sobre una placa. Posee un regulador

de voltaje de 3.3V en placa, esto permite alimentar la placa directamente del puerto micro-USB o por los pines 5V y GND. Los pines de entradas/salidas (GPIO) trabajan a 3.3V por lo que para la conexión a sistemas de 5V es necesario utilizar conversores de nivel. El SoM(System on Module) ESP-12E fabricado por Ai-Thinker integra en un módulo el SoC ESP8266, memoria FLASH, cristal oscilador y antena WiFi en PCB [44].

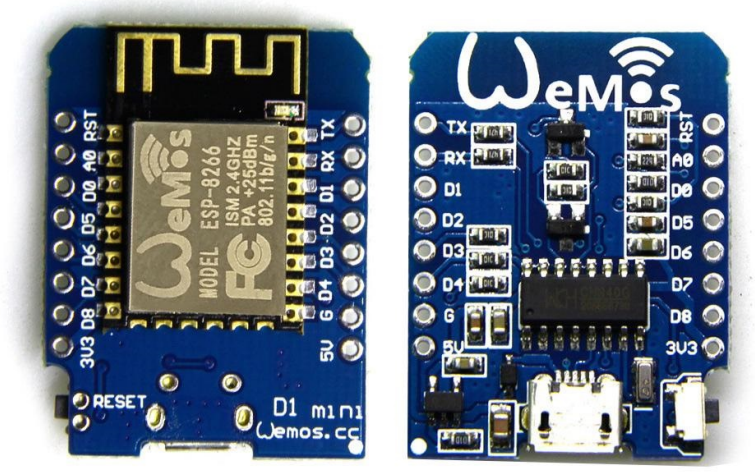


Figura 3.7: Dispositivo Wemos D1 mini ESP8266

La plataforma ESP8266 permite el desarrollo de aplicaciones en diferentes lenguajes como: Arduino, Lua, MicroPython, C/C++, Scratch. Al trabajar dentro del entorno Arduino provee un lenguaje de programación conocido y un Integrated Development Environment (IDE) sencillo de utilizar, además adquiere información sobre proyectos y librerías disponibles en Internet. Las características de este dispositivo se encuentran detalladas en el apéndice B.

3.4. Raspberry Pi

Raspberry Pi es un Single Board Computer (SBC) desarrollado en Reino Unido por la fundación Raspberry Pi, destinado para la enseñanza y promoción de ciencia básica de computación en escuelas y colegios de Reino Unido [45]. Raspberry Pi cuenta con un microcontrolador basado en ARM. Además, la placa Raspberry Pi funciona con una tarjeta SD preparada con el sistema operativo, un teclado USB, un mouse, una pantalla y una fuente de alimentación, por lo que trabaja como un ordenador [46]. Adicionalmente, si se configura adecuadamente, es posible administrar el dispositivo de manera remota desde otro ordenador en la misma red a través de Secure Shell (SSH) o Remote Desktop Protocol (RDP), evitando la tediosa necesidad de usar periféricos [47].

El procesamiento de datos y tratamiento de la información es una etapa importante en el desarrollo de un dispositivo de monitoreo [48]. Para este propósito se emplea una placa de desarrollo que es un *hardware* utilizado frecuentemente en proyectos electrónicos. De acuerdo con [49], [50], [51] la tarjeta Raspberry Pi es una placa que resulta muy útil al realizar proyectos debido a su bajo costo, bajo consumo energético y pequeño tamaño que permite hacer portables las soluciones. Además, la posibilidad de incorporar sensores, permite el despliegue de una gran cantidad de aplicaciones y al ser una plataforma que está orientada a la educación cuenta con una amplia comunidad que aporta información fiable relacionada con su manejo. En la figura 3.8 se aprecia el modelo Raspberry Pi 3B+ usado para este proyecto, además en el apéndice B se detallan sus principales características.

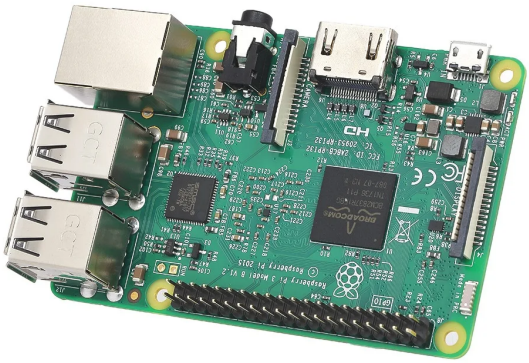


Figura 3.8: Raspberry Pi 3 modelo B+

3.4.1. Python

Python es un lenguaje de programación interpretado con una sintaxis limpia y que favorece un código legible, similar al pseudocódigo utilizado para esquematizar la programación [45]. Las características que marcan a Python son:

- Ser tanto una herramienta potente como intuitiva.
- Que cualquiera pueda contribuir a su desarrollo (Open source).
- Muy legible y entendible.
- Que sea de apropiado uso para tareas diarias, es decir, que permita tiempos cortos de desarrollo.

Python es una herramienta que permite realizar una programación completa y por ello se le denomina lenguaje de programación multiparadigma, ya que permite tanto programación con diseño orientado a objetos, unidades de testeo, generación de documentación e incluso alguna interacción con el sistema operativo. Actualmente

Python tiene un amplio conjunto de librerías que extiende su funcionalidad en el ámbito científico, incluyendo tareas tanto de tratamiento de datos, como visualización, cálculo numérico y simbólico entre otras aplicaciones específicas. Todo ello respaldado por una comunidad de usuarios que, debido a la filosofía código abierto, están dispuestos a compartir su código y mejorarlo entre todos [45].

Python viene instalado por defecto en la placa Raspberry Pi y se usa para generar *scripts* o programas; este es el lenguaje central en Raspberry Pi OS [51]. Se usa en gran cantidad de proyectos integrados con Raspberry Pi como [52] en donde se emplea para recoger datos de sensores de temperatura, presión, humedad para monitorear el estado de una embarcación, o como [53] en donde se emplea Python en conjunto con Raspberry Pi para implementar un sistema de automatización domótico, que incluye reconocimiento facial. En el caso de [54] se diseña una aplicación en Python sobre Raspberry Pi para recabar información de una red de sensores inalámbricos.

3.5. Componentes LAMP

LAMP hace referencia a un conjunto de subsistemas de *software* necesarios para alcanzar una solución general como sitios web o servidores dinámicos [48]. Los servidores web basados en Linux constan de cuatro componentes de software como indica la figura 3.9. Estos componentes, dispuestos en capas forman la pila de software. Los sitios web y las aplicaciones web se ejecutan sobre esta pila subyacente [55].

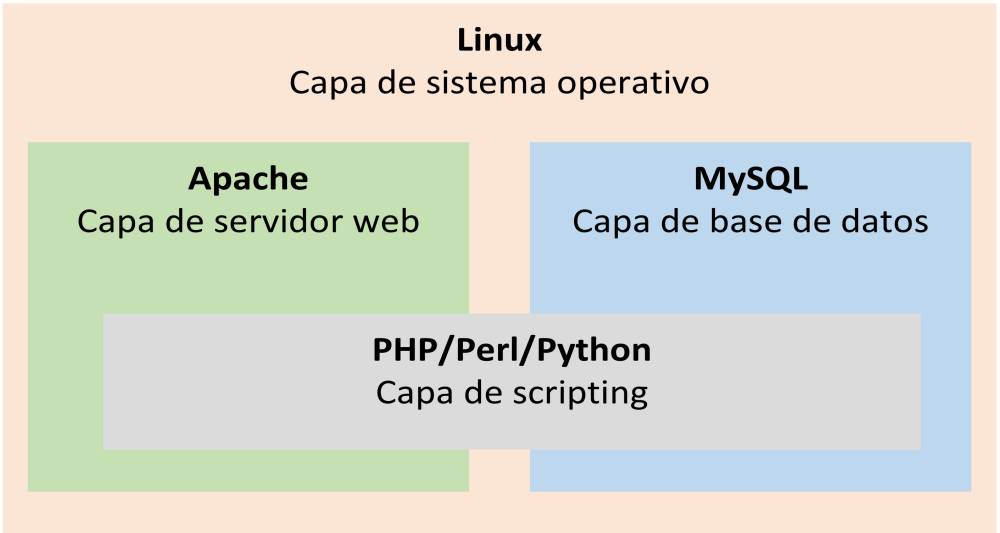


Figura 3.9: Modelo de capas de LAMP

Los componentes de software comunes que componen una pila LAMP tradicional son:

3.5.1. Linux

El sistema operativo (SO) constituye la primera capa. Linux sienta las bases para el modelo de pila. Todas las demás capas se ejecutan encima de esta capa.

3.5.2. Apache

La segunda capa consta de un software de servidor web, generalmente Apache Web Server. Esta capa reside encima de la capa de Linux. Un servidor web recibe peticiones de clientes y responde con el envío de ficheros, texto plano (html, php) o binarios (gif, jpeg). Constantemente escucha las peticiones de conexión de los clientes en determinados puertos: 80 para HTTP, 443 para HTTPS [56].

3.5.3. MySQL

La tercera capa corresponde a las bases de datos. MySQL almacena detalles que se pueden consultar mediante secuencias de comandos para construir un sitio web. MySQL generalmente se encuentra en la parte superior de la capa de Linux junto con Apache.

3.5.4. Perl, PHP o Python

En la parte superior está la cuarta y última capa. La capa de secuencias de comandos consta de PHP y/u otros lenguajes de programación web similares. Los sitios web y las aplicaciones web se ejecutan dentro de esta capa.

3.6. MQTT

Message Queuing Telemetry Transport (MQTT) es un protocolo de mensajería de intercambio que utiliza el estándar de publicación y suscripción para transportar mensajes entre un servidor y los clientes. Se ejecuta sobre TCP/IP y se puede ejecutar en protocolos de red que proporcionan conexiones ordenadas, sin pérdidas y bidireccionales. Está estandarizado por un comité técnico de la Organización para el Avance de los Estándares de Información Estructurada. MQTT se convierte en un buen candidato para usarse en contextos de IoT y comunicación de máquina a máquina (IoT/M2M), posee las siguientes características [57]:

- Diseñado para ser liviano.

- Abierto y fácil de implementar, especialmente en contextos donde Internet puede ser costoso.
- Consume poco ancho de banda.
- Útil en sistemas con recursos de memoria o procesamiento limitados.

El protocolo MQTT generalmente se usa en sistemas embebidos porque puede hacer que los sensores se comuniquen con el sistema *backend* y es fácil de implementarlo [58]. Cualquier conexión MQTT generalmente involucra dos tipos de agentes: *broker* o servidor y clientes MQTT [59].

Sus características lo convierten en un protocolo muy usado en gran variedad de proyectos, tal es el caso de [60], donde se emplea una Raspberry pi como broker MQTT donde el microcontrolador ESP32 (cliente MQTT) publica datos en un sistema de monitoreo diseñado para agricultores urbanos. En [61] en cambio el autor analiza el desempeño de un sistema de monitoreo de sensores usando Raspberry Pi a través del protocolo MQTT, en este caso la Raspberry esta conectada a sensores y tiene el rol de cliente MQTT que publica los datos en un servidor en la nube. Una implementación de bajo costo de MQTT usando ESP8266 la presenta [41], el dispositivo ESP8266 funciona como cliente MQTT, recoge datos de sensores y los publica en un adafruit.io que constituye el broker MQTT que monitorea el sistema implementado. Por otra parte [62] diseña un sistema inalámbrico de monitoreo de frecuencia cardíaca usando MQTT como protocolo para envío de datos entre el módulo Wi-Fi ESP8266 y una Raspberry Pi que actúan como cliente y broker respectivamente.

3.6.1. Broker MQTT

Un broker MQTT es un dispositivo central encargado del manejo de la comunicación entre los clientes MQTT y la distribución de mensajes entre ellos [63]. Un broker puede manejar hasta miles de clientes MQTT conectados al mismo tiempo. Cuando se recibe un mensaje, el broker encuentra todos los clientes que tienen una suscripción al tema recibido [64]. Además, un broker cumple tareas de autenticación y autorización de los clientes por motivos de seguridad. Para la comunicación cifrada entre el broker y los clientes, se utiliza el cifrado Transport Layer Security (TLS) y Secure Sockets Layer (SSL), el mismo protocolo de seguridad que utiliza el protocolo Hypertext Transfer Protocol (HTTP) [65]. Hay varios brokers de mensajes empleados con el protocolo MQTT como Mosquito, broker MQTT v3.1/v3.1.1 de código abierto y HiveMQ, un broker MQTT empresarial [64].

3.6.2. Cliente MQTT

Cualquier objeto IoT es un cliente MQTT que envía o recibe datos de telemetría [64]. El cliente es un dispositivo IoT, aplicación web, aplicación móvil entre otros. Los que publican un mensaje al broker con un tema se denominan *publishers* y los que suscriben uno o más temas para leer mensajes específicos son *suscribers*. Los *suscribers* pueden recibir mensajes de varios *publishers* y enviarlos a otros *suscribers*. Todos los clientes establecen la conexión con el broker. Los *publishers* desconocen el destino de los mensajes enviados y los *suscribers* desconocen el origen de los mensajes recibidos [66].

Para crear un cliente MQTT desde un dispositivo, se debe instalar una biblioteca MQTT y establecer una conexión a un agente MQTT a través de cualquier tipo de red. Las bibliotecas MQTT están disponibles para diferentes tipos de lenguajes y plataformas de programación, por ejemplo, Python, JavaScript, PHP, C, C++, Android, iOS, etc.

3.6.3. Mosquitto

Eclipse Mosquitto es un intermediario de mensajes de código abierto (con licencia EPL/EDL) que implementa las versiones 5.0, 3.1.1 y 3.1 del protocolo MQTT. Mosquitto es liviano y adecuado para usarse en todos los dispositivos, desde computadoras de placa única de bajo consumo hasta servidores completos. El proyecto Mosquitto también proporciona una biblioteca C para implementar clientes MQTT y los muy populares clientes MQTT de línea de comandos mosquitto_pub y mosquitto_sub [67].

3.7. Algoritmo de clasificación

Es necesario implementar un algoritmo de clasificación basado en aprendizaje supervisado para discernir a qué posturas corresponden los datos obtenidos. Algunos algoritmos utilizados por [68], [69] y [70] en trabajos relacionados son el k-Nearest Neighbor (k-NN), Support vector machine (SVM), Naive Bayes, Random Forest. Los resultados de estos autores indican que estos algoritmos funcionan adecuadamente con tasas de rendimiento altas. Sin embargo, para este proyecto el elegido es k-NN, ya que según lo mencionado en los artículos previos, este presenta un mejor balance entre la efectividad de clasificación y tiempo computacional requerido para el procesamiento, además de que se recomienda para plataformas de recursos limitados, en este caso la Raspberry Pi.

k-NN a partir de un conjunto de datos inicial clasifica todos los datos nuevos. El algoritmo coloca cada dato nuevo en el grupo que corresponda, según tenga K vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes y ordena dichas distancias de menor a mayor para seleccionar el grupo al que pertenecen. La asignación de la clase de un nuevo dato depende del número K, es por ello que el algoritmo primero escoge el valor K y su distancia. Después, encuentra el K vecino más cercano del conjunto de datos a clasificar. Por último, asigna la etiqueta de clase por "votación mayoritaria" [69].

Para determinar dicha distancia, el algoritmo emplea la distancia euclidiana entre una muestra de prueba y las muestras de entrenamiento especificadas. Sea x_i una muestra de entrada con p características $(x_{i1}, x_{i2}, ..., x_{ip})$, h el número total de muestras de entrada $(i = 1, 2, ..., h)$ y p el número total de características $(j = 1, 2, ..., p)$. La distancia euclidiana entre la muestra x_i y x_l $(l = 1, 2, ..., h)$ es:

$$d(x_i, x_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + ... + (x_{ip} - x_{lp})^2} \tag{3.5}$$

La ventaja de usar el algoritmo k-NN es que no hace ninguna suposición sobre los datos. Clasifica datos complejos utilizando funciones sencillas como aproximaciones locales.

3.8. Cambios posturales en la prevención de UPP

Para la prevención de las úlceras por presión de acuerdo a lo mencionado en [5], [2] es importante realizar cambios posturales cada 2-3 horas durante el día y cada 4 horas máximo en la noche. Además, únicamente llevar a cabo los cambios posturales cuando no existan contraindicaciones. Las posturas mas recurrentes según [71] corresponden a decúbito supino, decúbito lateral derecho y decúbito lateral izquierdo. En [12], [14] analizan el comportamiento de un sistema para la prevención de UPP considerando las posturas mencionadas y la postura de sedestación (sentado). La figura 3.10 obtenida de [5] y [15] muestra una rutina con los cambios posturales sugeridos a lo largo del día.

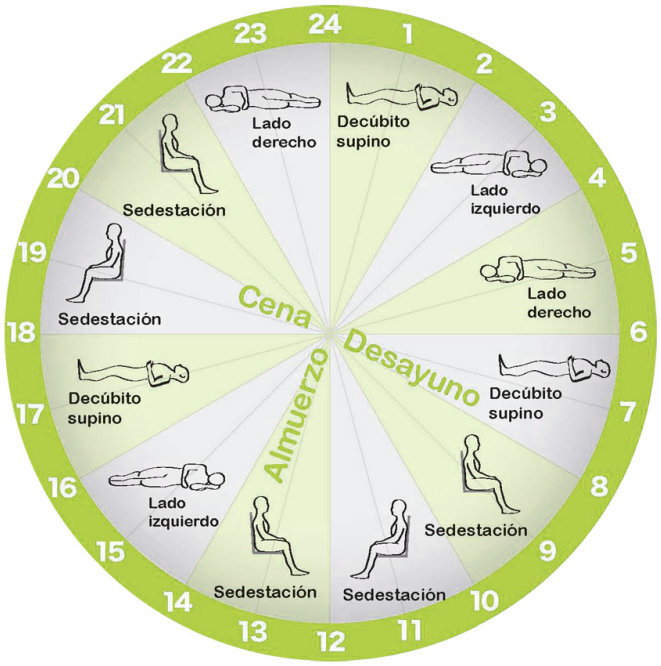


Figura 3.10: Rutina con cambios posturales a lo largo del día

Es recomendable usar almohadas para evitar el contacto entre prominencias óseas, esto ayuda a disminuir la presión en distintas áreas, según [2] y [71]. El estudio de clasificación de [14] considera las posturas mencionadas para la evaluación del rendimiento de un sistema de prevención de UPP. La figura 3.11 tomada de este trabajo señala el uso de almohadas como buenas prácticas para el cuidado de pacientes.

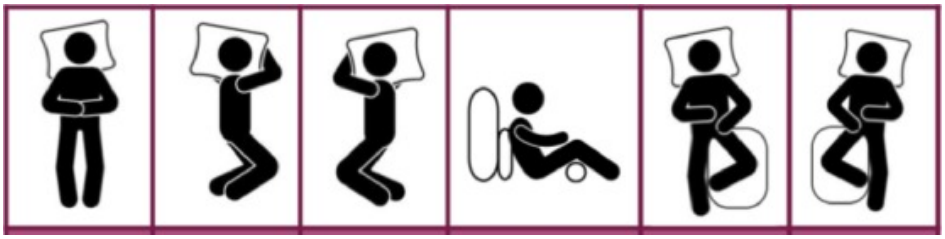


Figura 3.11: Buenas prácticas con almohadas para aliviar los puntos de presión y prevenir UPP

La postura decúbito ventral la considera [72] en su trabajo, como parte del plan de rotación de un paciente para evitar la generación de úlceras, sin embargo esta postura no la consideran en la mayoría de casos. Los cambios posturales identificados como prioritarios para la prevención de UPP son:

- Decúbito Supino (boca arriba).
- Decúbito lateral derecho (de costado derecho).

- Decúbito lateral derecho (de costado izquierdo).
- Sedestación (sentado completamente).
- Semisedestación (sentado con una ligera inclinación hacia atrás).

3.9. Escala de Braden

La escala de Braden fue desarrollada en 1985 en Estados Unidos, en el contexto de un proyecto de investigación en centros sociosanitarios, como intento de dar respuesta a algunas de las limitaciones de la escala de Norton, realizando su escala a través de un esquema conceptual en el que reseñaron, ordenaron y relacionaron los conocimientos existentes sobre UPP. La escala de Braden consta de 6 subescalas: percepción sensorial, exposición de la piel a la humedad, actividad física, movilidad, nutrición y peligro de lesiones cutáneas por fuerzas de fricción y/o cizalla. Es una escala negativa, es decir, que a menor puntuación el riesgo es más elevado, con un rango que oscila entre los 6 y los 23 puntos. Se consideran pacientes de riesgo los que obtienen puntuaciones iguales o inferiores a 18, y se clasifican como: pacientes de muy alto riesgo (puntuación ≤ 9), de alto riesgo (puntuaciones entre 10 y 12), de riesgo moderado (puntuaciones entre 13 y 14) y de bajo riesgo (puntuaciones entre 15 y 18) [73].

4. Diseño e implementación del sistema

Este capítulo presenta una descripción general de la metodología empleada para el desarrollo del proyecto, así como los procedimientos y actividades llevadas a cabo a través de la metodología definida para cumplir los objetivos planteados.

4.1. Diagrama conceptual del sistema

De manera general el sistema está diseñado en una arquitectura de tres niveles, según muestra el esquema de la figura 4.1. A continuación se indican las características más relevantes de cada etapa.

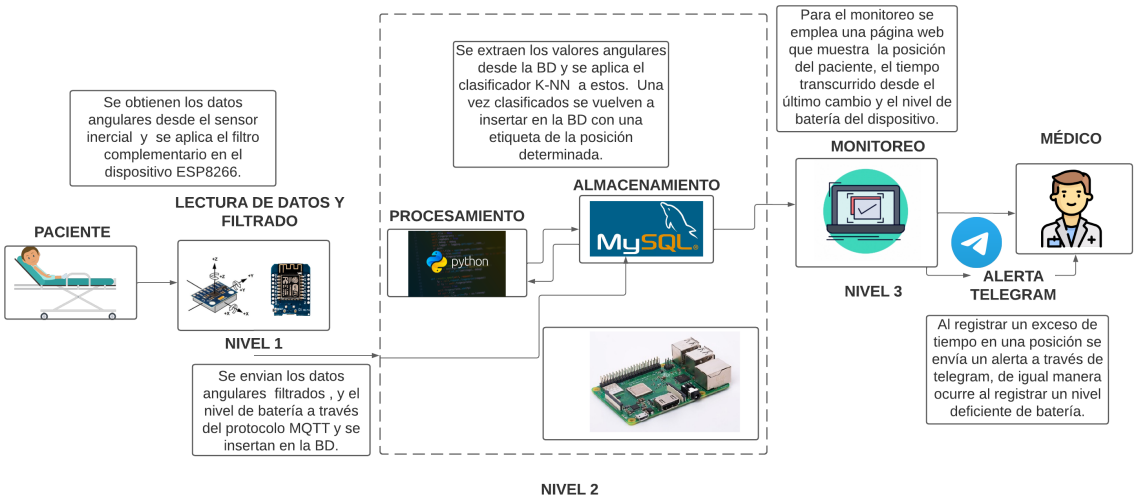


Figura 4.1: Esquema general del proyecto

4.1.1. Nivel 1

El nivel 1 comprende la adquisición de datos angulares desde la IMU, estos datos son interpretados por el D1 Mini ESP8266. Los valores medidos corresponden a los ángulos Pitch y Roll. Para que estos datos sean confiables es necesario realizar un proceso de calibración. Además en esta etapa es necesario aplicar un filtro a las señales obtenidas de la IMU, con el fin de reducir ruido y retardo en la medición.

4.1.2. Nivel 2

Este nivel corresponde al procesamiento de los datos obtenidos previamente. Los datos filtrados son enviados desde el D1 mini ESP8266 mediante el protocolo MQTT hacia la Raspberry Pi, que a su vez inserta estos registros en la base de Datos MySQL. Posterior a esto, el algoritmo clasificador, determina la posición del paciente. Finalmente, una función de python guarda la posición determinada previamente en la base de datos.

4.1.3. Nivel 3

El nivel 3 consiste en extraer la posición del paciente y nivel de batería del dispositivo desde la base de datos para presentarlo en la aplicación web elaborada para monitorizar al paciente. Los parámetros incluidos en este monitor son el tiempo que permanece un paciente en alguna posición y el nivel de carga del dispositivo. De manera adicional, en esta etapa se implementa un sistema de alerta a través de telegram.

4.2. Adquisición, procesamiento, almacenamiento y monitoreo de información

Esta sección muestra el diseño y desarrollo de los dos primeros niveles definidos anteriormente para el correcto desenvolvimiento del sistema planteado.

4.2.1. Captura y acondicionamiento

La captura consiste en adquirir datos a través de los sensores que vienen incorporados en la IMU MPU6050. Se emplea la placa D1 Mini ESP8266 para poner a punto los sensores y capturar los datos que estos arrojen. Estos dispositivos se comunican a través del estándar I2C por puertos específicos. En cuanto al diseño del filtro se parte por la ecuación planteada en capítulos anteriores, es decir:

$$\theta_{estimado} = \alpha \cdot (\theta_{est\_previo} + \omega_{gyro} \cdot dt) + (1 - \alpha) \cdot \theta_{acel} \tag{4.1}$$

El valor a determinar para el correcto funcionamiento del filtro es el factor de fidelidad α. Este se lo obtiene luego de realizar las respectivas pruebas, tal y como se muestra en la figura 4.2. El valor que arroja mejores resultados es $\alpha = 0{,}98$, por lo que la ecuación implementada en el código de arduino es:

$$\theta_{estimado} = 0{,}98 \cdot (\theta_{est\_previo} + \omega_{gyro} \cdot dt) + (0{,}02) \cdot \theta_{acel} \tag{4.2}$$

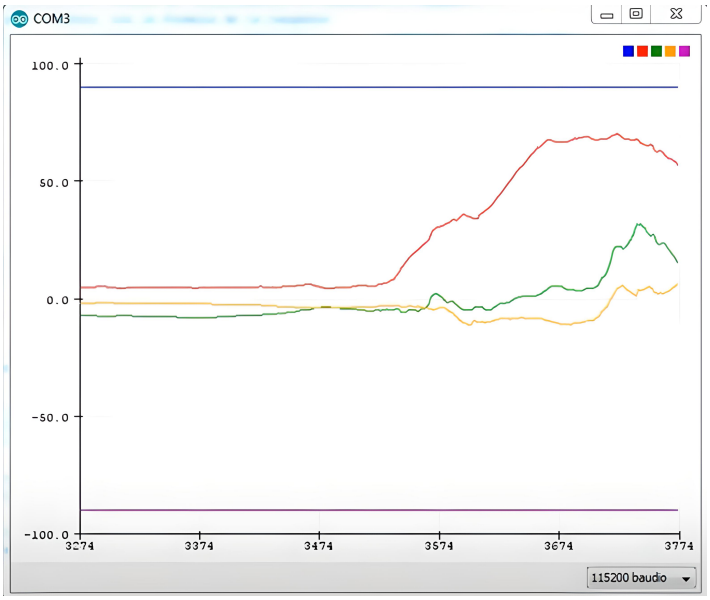


Figura 4.2: Valores angulares luego de pasar por el filtro complementario

Por otro lado, uno de los datos capturados es el voltaje de la batería a través del puerto analógico de la placa D1 Mini ESP8266, a fin de obtener una estimación del porcentaje de carga disponible. Para esto se considera que la tensión de la batería es proporcional a su capacidad disponible, es decir, cuando hay consumo, la corriente sale de la batería y por lo tanto la tensión de la batería baja. Esta bajada es proporcional a la tasa de corriente de salida. Esta técnica es usualmente usada en sistemas off-grid [74]. La figura 4.3 muestra los componentes que interactúan en esta etapa.

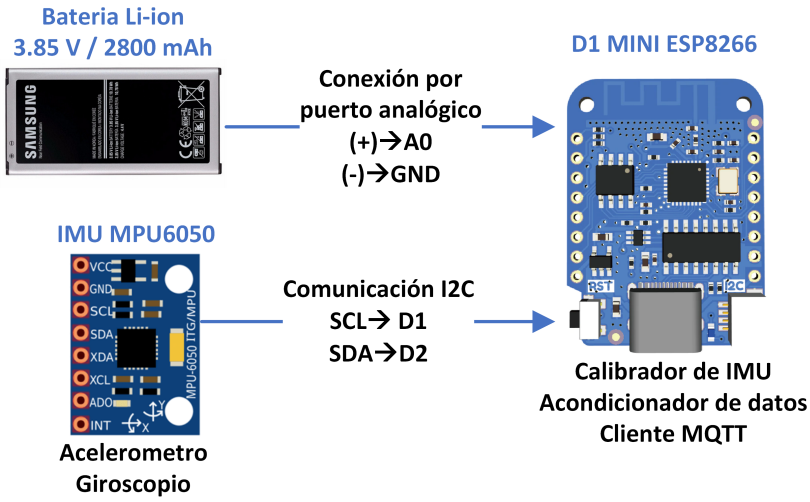


Figura 4.3: Interacción de componentes en la captura y acondicionamiento de datos

En el acondicionamiento se opera sobre los datos capturados para obtener las magnitudes deseadas, tal es el caso de los ángulos roll y pitch, así como el voltaje de la batería. En esta etapa, se crea y carga el script de Arduino IDE $D1\_Mini\_ESP8266.ino$ en la placa D1 Mini ESP8266 para que el microcontrolador lleve a cabo el trabajo de captura y acondicionamiento. El detalle de este script está en el apéndice C.1. A través de este script se conecta la placa D1 Mini ESP8266 a la red Wireless Fidelity (Wifi). A fin de corregir cualquier desnivel de los sensores de la IMU MPU6050 se calibran offsets. Luego, para transmitir los datos acondicionados se configura un cliente MQTT, este cliente publica los datos deseados hacia un broker, en el contexto del proyecto corresponde a la Raspberry Pi. Finalmente, a partir de cada dato capturado por la IMU se calculan los valores requeridos y se los transmite.

A través del monitor serie de Arduino IDE se visualiza la salida del script desarrollado (véase figura 4.4), de esta manera se lleva un control del correcto funcionamiento de los dispositivos.

```
Conectando a ssid: Red_Tesis
.......
Conectado a red WiFi!
Dirección IP:
192.168.2.101
Sensor iniciado correctamente
Calibrando, no mover IMU
promedio:       964     301     14217   -468    -146    70
promedio:       1134    363     17027   -463    -147    68

promedio:       13      2       16693   -3      7       -6
promedio:       9       -7      16711   3       -3      -2
Intentando conexión Mqtt...Conectado!
```

Figura 4.4: Salida del script $D1\_Mini\_ESP8266$ por monitor serie de Arduino IDE

4.2.2. Recepción y almacenamiento de datos vía MQTT

Los datos enviados por la placa D1 Mini ESP8266 son recibidos en la Raspberry Pi a través del protocolo MQTT. Estos datos son almacenados en una base de datos llamada TESIS que esta levantada en la misma Raspberry Pi. En la figura 4.5 se presenta una diagrama con la interacción de los componentes indicados.

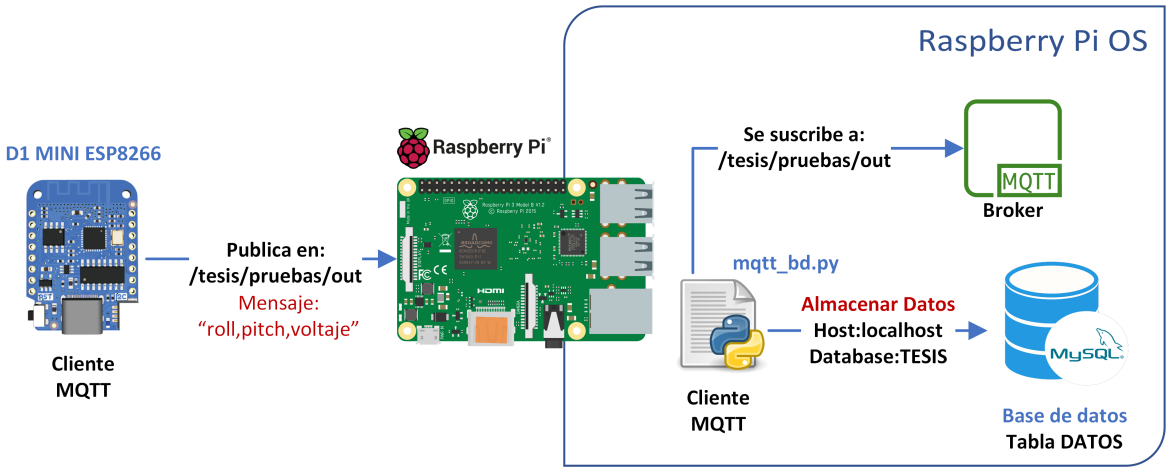


Figura 4.5: Interacción de componentes en la recepción y almacenamiento de datos en Raspberry Pi

La placa D1 Mini ESP8266 (cliente MQTT) publica los datos en el tema "tesis/-pruebas/out", estos datos llegan a la Raspberry Pi que constituye el broker. El script *mqtt_bd.py* desarrollado en Python permite crear un cliente MQTT que se suscribe al tema ya mencionado para recopilar los datos que se están publicando. Una vez obtenidos los datos, se crea una conexión a la base de datos TESIS para almacenar los datos recibidos en la tabla DATOS. El dato almacenado contiene tres campos que son ID, FECHA y MEDICION. Verificar el apéndice C.2 para mas información del script desarrollado.

Una vez ejecutado el script, a través de la consola de Raspberry Pi aparecen los datos que esta recolectando el cliente MQTT. Una evidencia de esta ejecución aparece en la figura 4.6.

```
pi@raspberrypi:~/Documents $ python mqtt_bd.py
connected (b'esp8266-sub')
------------------------------
payload: b'-0.49,18.11,4.11'
------------------------------
payload: b'-1.19,87.67,4.11'
------------------------------
payload: b'-0.92,-0.02,4.11'
------------------------------
payload: b'50.28,8.14,4.11'
------------------------------
payload: b'71.26,5.85,4.11'
------------------------------
payload: b'1.36,-78.91,4.11'
------------------------------
payload: b'-0.21,-1.02,4.11'
------------------------------
payload: b'-0.05,-0.01,4.11'
```

Figura 4.6: Salida del script *mqtt_bd.py* a través de la consola de Raspberry Pi

La figura 4.7 muestra los registros obtenidos de la tabla DATOS al ingresar a través de phpmyadmin a la base de datos. Los registros coinciden con los valores de la figura 4.6. Esto indica que los datos recibidos efectivamente se están almacenando.

Figura 4.7: Registros de la tabla DATOS visualizados en phpmyadmin

4.2.3. Clasificación de datos

El procesamiento de los datos almacenados consiste en la implementación del algoritmo de clasificación. Para llevar esto a cabo, como se indica en la figura 4.8 interactúan entre sí la base de datos TESIS y un script *clasificador.py* desarrollado en python.

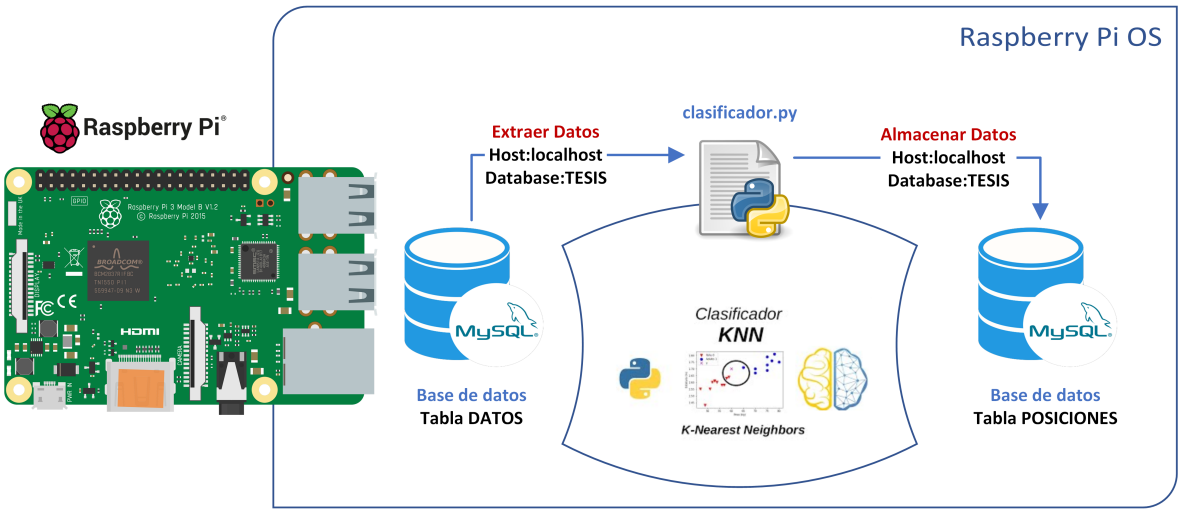


Figura 4.8: Interacción de componentes en la clasificación de datos

El script crea una conexión con la base de datos TESIS, específicamente la tabla DATOS, de aquí extrae el campo MEDICION que contiene el dato con valores angulares a clasificar. El dato recuperado es almacenado en una variable y pasa a través del algoritmo de clasificación k-NN implementado. En la figura 4.9 se presenta un diagrama de flujo que indica el funcionamiento del algoritmo de clasificación.

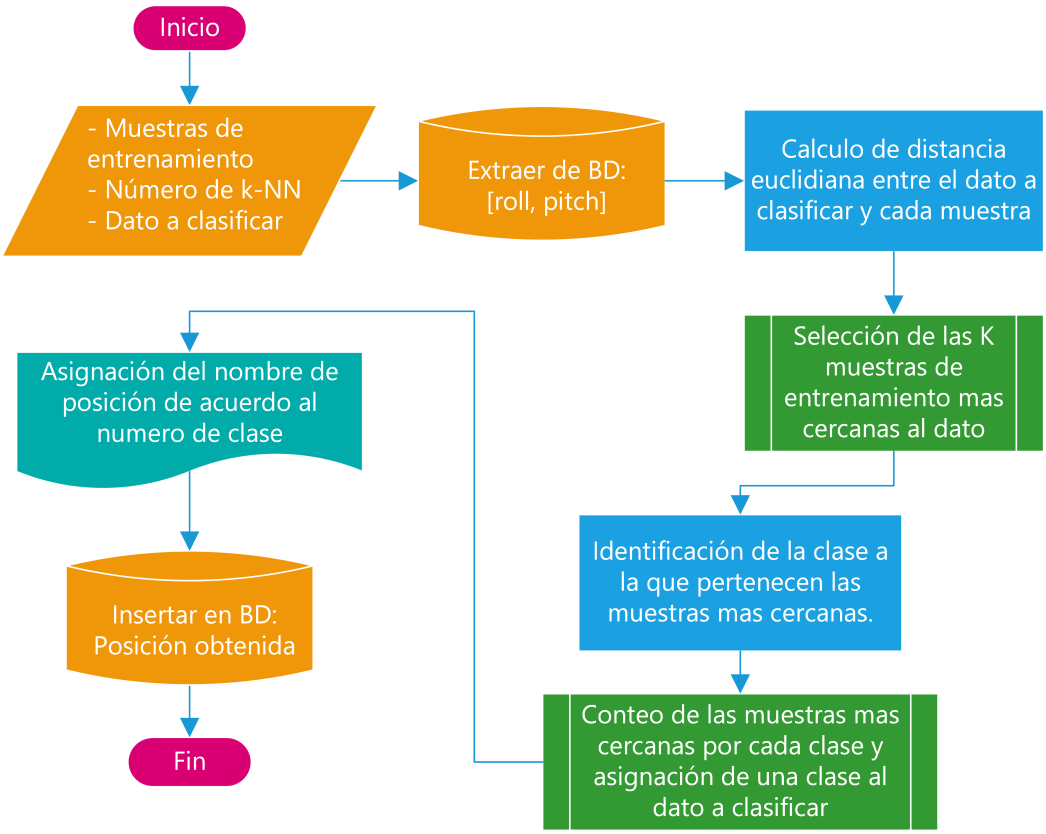


Figura 4.9: Diagrama de flujo del algoritmo de clasificación

Para el funcionamiento del algoritmo, en el script se definen algunas muestras de entrenamiento obtenidas con anterioridad por cada posición a clasificar. A la salida del algoritmo de clasificación se obtiene un valor que indica la posición a la que pertenece el dato clasificado. Este valor se almacena en la base de datos TESIS en la tabla POSICIONES. El registro almacenado contiene tres campos, estos son, ID, FECHA y POSICION. El apéndice C.3 contiene el detalle del script desarrollado.

Al ejecutar el script, a través de la consola de Raspberry Pi se visualiza un array con los valores angulares a clasificar y el voltaje de batería restante. Además, un texto que indica a que clase pertenece el dato clasificado. El número de clase y su respectiva posición se resume a continuación:

- Clase 0: Decúbito Supino.
- Clase 1: Decúbito Lateral Izquierdo.
- Clase 2: Decúbito Lateral Derecho.
- Clase 3: Sedestación.
- Clase 4: Semisedestación.

La figura 4.10 muestra el resultado de ejecución del script, este corresponde a los datos recolectados que fueron presentados en la figura 4.6.

```
pi@raspberrypi:~/Documents $ python clasificador.py
['-0.49', '18.11', '4.11']
El valor pertenece a la clase 0
['-1.19', '87.67', '4.11']
El valor pertenece a la clase 1
['-0.92', '-0.02', '4.11']
El valor pertenece a la clase 0
['50.28', '8.14', '4.11']
El valor pertenece a la clase 3
['71.26', '5.85', '4.11']
El valor pertenece a la clase 3
['1.36', '-78.91', '4.11']
El valor pertenece a la clase 2
['-0.21', '-1.02', '4.11']
El valor pertenece a la clase 0
['-0.05', '-0.01', '4.11']
El valor pertenece a la clase 0
```

Figura 4.10: Salida del script *clasificador.py* a través de la consola de Raspberry Pi

4.3. Aplicación web para monitoreo y envío de alertas

Esta sección constituye la implementación del tercer nivel de la arquitectura presentada en el diseño del proyecto. Aquí se diseña la aplicación web para monitorización de pacientes. La aplicación web está desarrollada con Hypertext Preprocessor (PHP), HyperText Markup Language (HTML) e incluye secciones con JavaScript. De manera general el funcionamiento de la aplicación está basado en el diagrama de la figura 4.11.

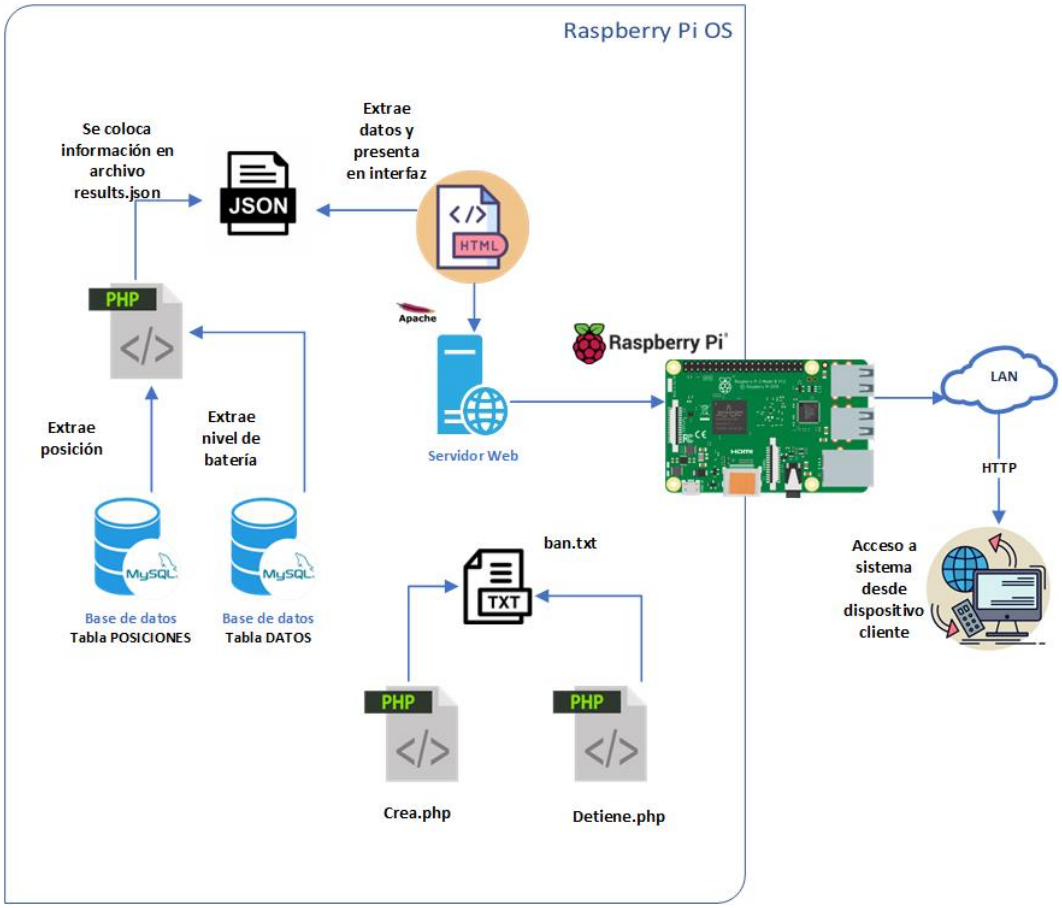


Figura 4.11: Interacción de componentes en el diseño de aplicación web

4.3.1. Desarrollo web en PHP

La aplicación web cuenta con tres scripts de código PHP indicados en la figura 4.11. El primero llamado *Contador.php* extrae información procesada desde la base de datos, como es la posición y voltaje restante de batería del dispositivo. Además, cumple la función de calcular el tiempo que un sujeto permanece en una posición, al localizar un cambio en los datos inicia nuevamente el contador. Estos datos son almacenados en un archivo JavaScript Object Notation (JSON), de donde se extraen posteriormente para presentarlos en la interfaz HTML.

El siguiente código es *Crea.php*, este genera un documento llamado *ban.txt* que sirve como punto de partida para que el bucle de lectura de datos pueda iniciarse. Por último está el código *Detiene.php*, que elimina el documento *ban.txt*, para detener el bucle de extracción de datos.

4.3.2. Interfaz con HTML

La interfaz consta de tres secciones como se aprecia en la figura 4.12. La primera corresponde a la ventana principal que contiene una carátula del sistema desarrollado con un menú que lleva a cualquiera de las dos secciones restantes, estas contienen el test de Braden y una ventana con la interfaz de monitoreo respectivamente.

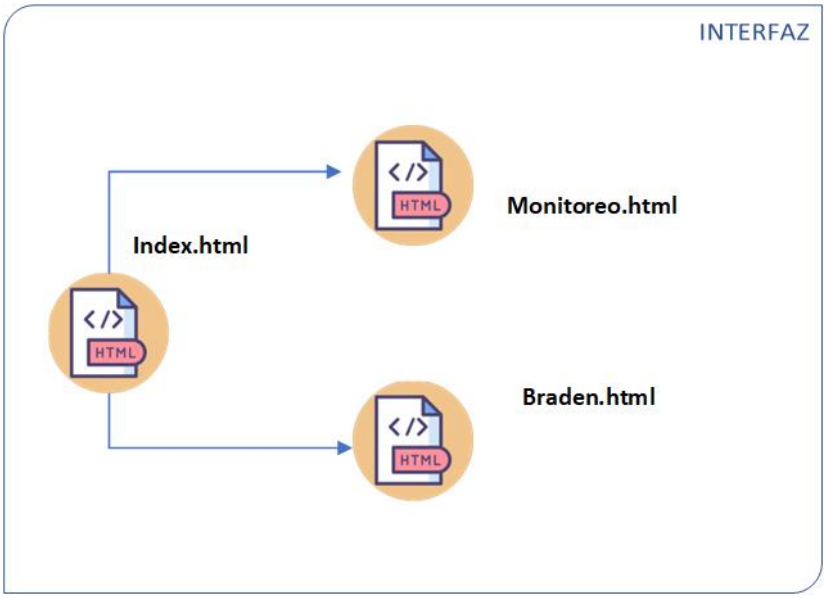


Figura 4.12: Scrips HTML en el desarrollo de la interfaz web

4.3.3. Configuración de alertas con Telegram

Se tienen dos tipos de alerta, la primera se activa cuando un paciente permanece por un tiempo excesivo en una misma posición, la segunda alerta se emite cuando el nivel de batería del dispositivo es demasiado bajo. El canal de comunicación para envío de alertas es Telegram. La figura 4.13 muestra el flujo de configuración de las alertas.

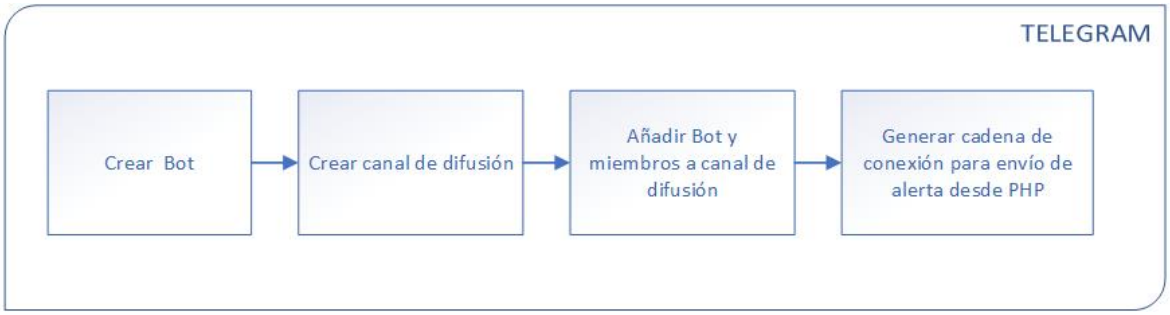


Figura 4.13: Flujo para la configuración de alertas con Telegram

4.4. Diseño y construcción del prototipo

El prototipo integra los componentes electrónicos utilizados en una tarjeta electrónica, ésta es encapsulada en un contenedor pequeño y compacto que protege la tarjeta, esto constituye el dispositivo para adquisición de datos. Además, comprende la elaboración de un artefacto que permite sujetar el dispositivo al cuerpo del paciente.

4.4.1. Tarjeta electrónica

El esquema del circuito electrónico se realizó en KiCad, la figura 4.14 muestra el diseño. En este circuito se integran las placas D1 Mini ESP8266 y la IMU MPU6050, además, un partidor de tensión para la medición del voltaje de la batería, un Light-Emitting Diode (LED) Red, Green, Blue (RGB) empleado como indicador del nivel de batería, un conector para la batería y un switch para el encendido/apagado del dispositivo.

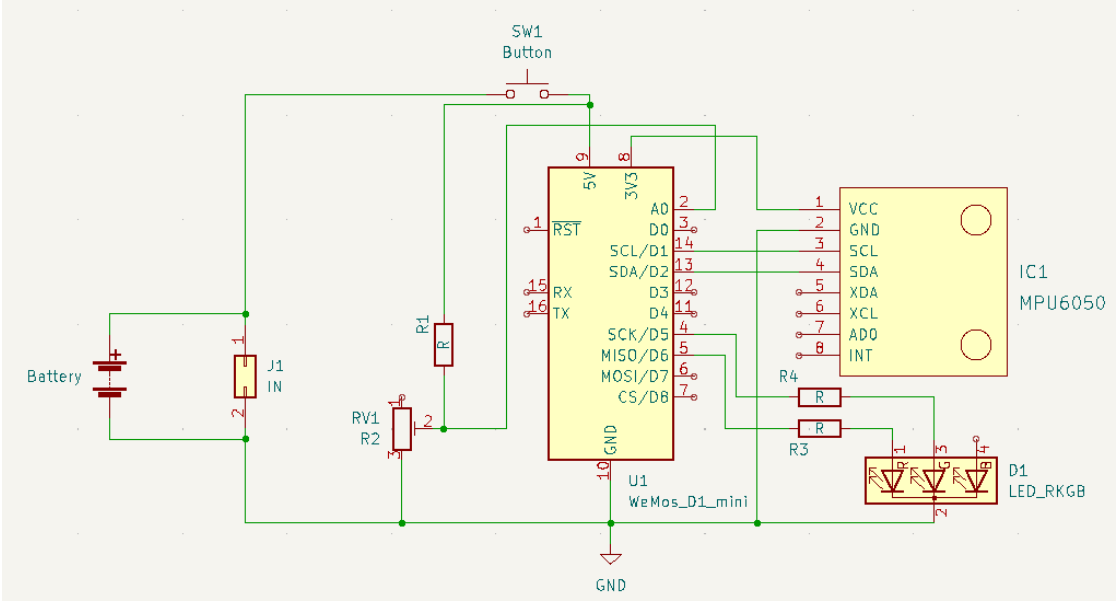


Figura 4.14: Esquemático de circuito electrónico

A partir del esquema, se genera la Printed Circuit Board (PCB) de igual manera en KiCad. La figura 4.15 muestra la PCB obtenida. La placa tiene un tamaño bastante reducido siendo sus dimensiones 5.67x3.82 cm.

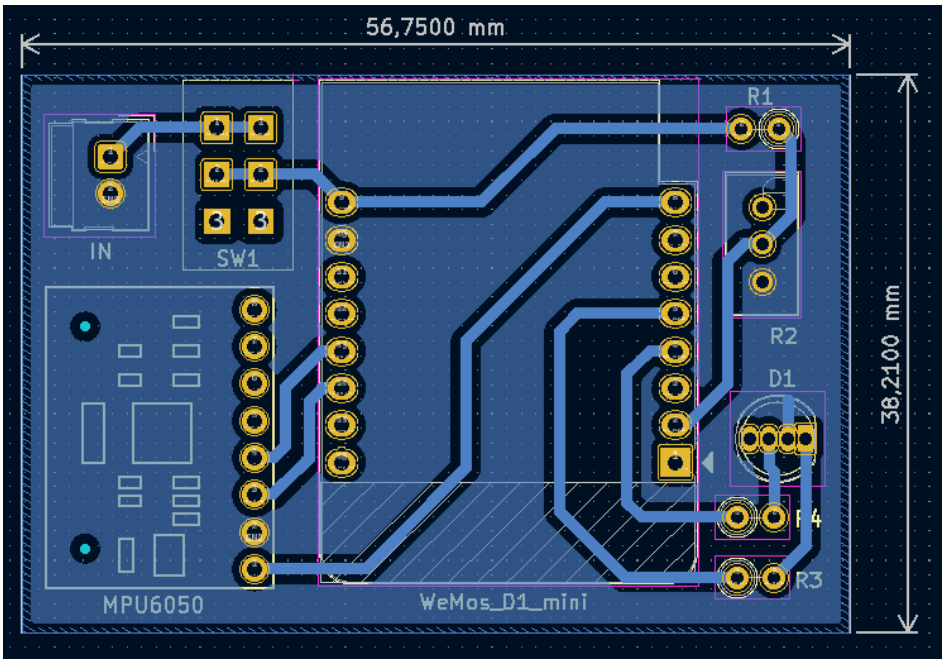
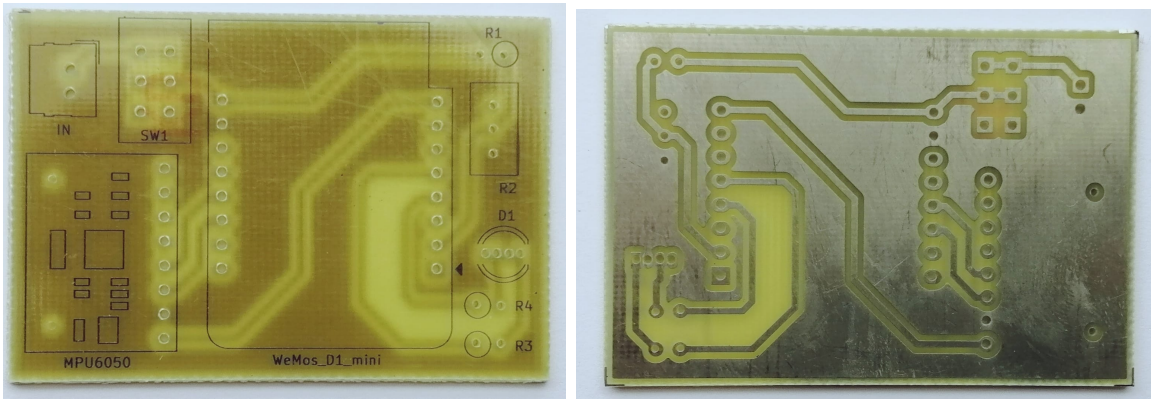


Figura 4.15: Diseño de placa de circuito impreso

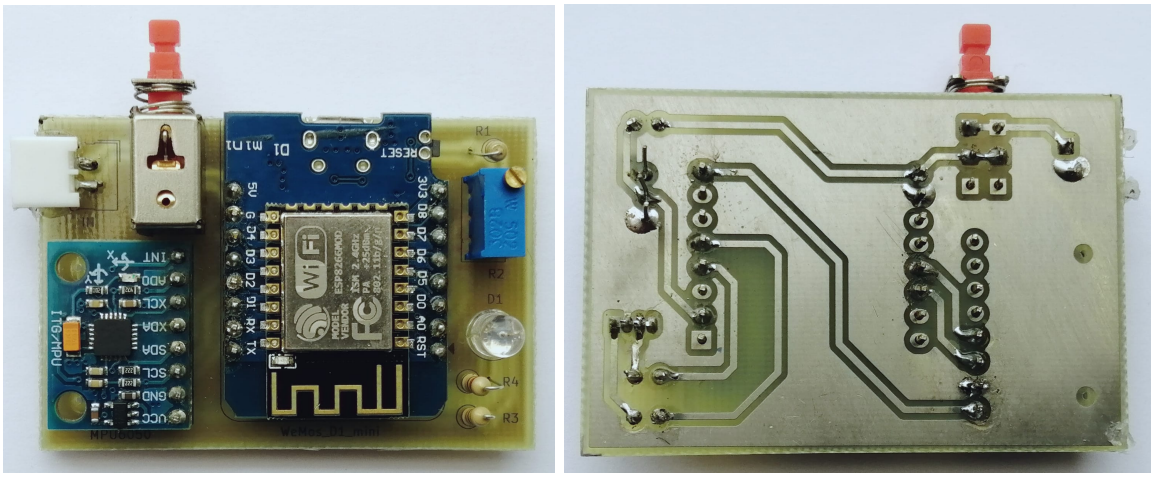
Con el diseño de la PCB se elabora la tarjeta electrónica, el resultado se muestra en la 4.16. En la cara frontal está impresa la máscara de componentes y en la cara trasera las pistas de cobre con los pads taladrados.



(a) Mascará de componentes (b) Pistas de cobre

Figura 4.16: Tarjeta electrónica elaborada

La figura 4.17 muestra la tarjeta electrónica elaborada en su totalidad con todos los componentes electrónicos montados y soldados.

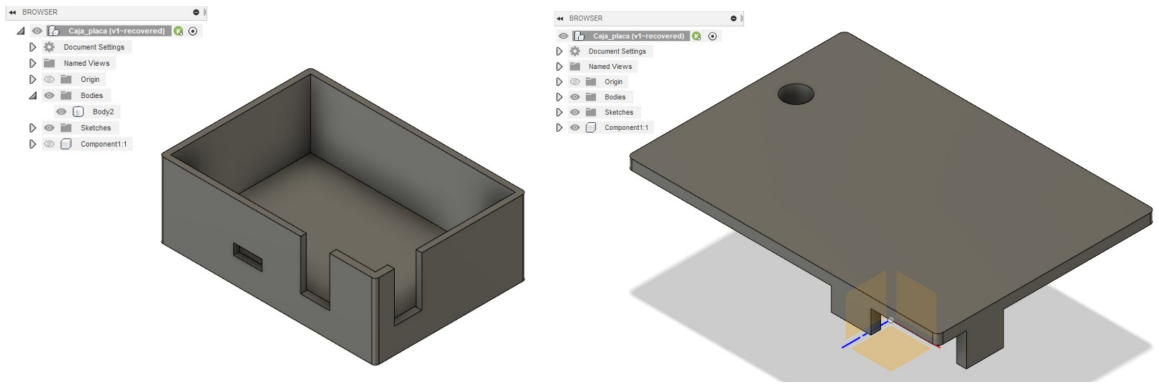


(a) Cara frontal (b) Cara trasera

Figura 4.17: Tarjeta electrónica con componentes electrónicos

4.4.2. Contenedor de tarjeta electrónica

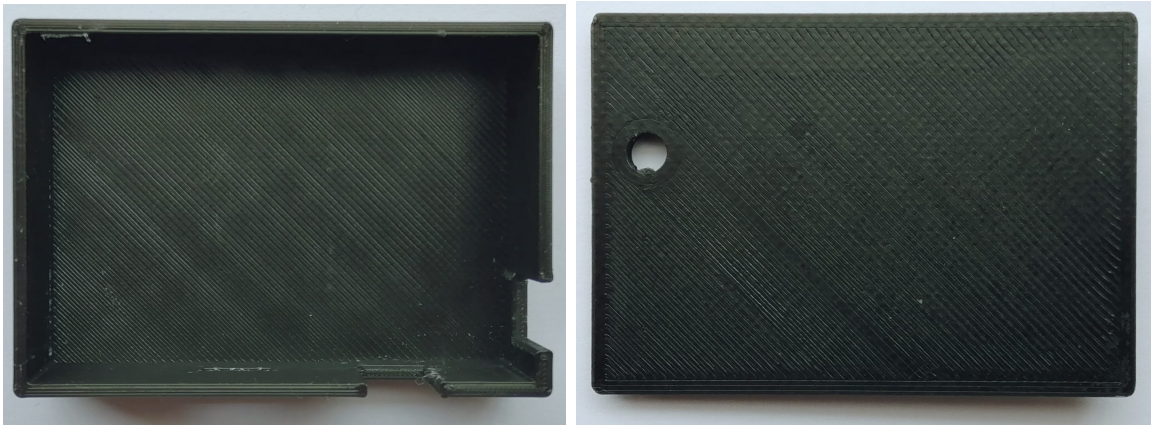
Un contenedor para la tarjeta electrónica es necesario ya que ésta debe estar fija y en lo posible aislada del contacto con los pacientes. Con esta consideración se elabora un contenedor plástico hecho a medida para ubicar la tarjeta electrónica. La figura 4.18 muestra el diseño 3D del contenedor, el mismo consta de una pieza que básicamente es una caja con un agujero y muescas para que la tarjeta embone de manera exacta y otra pieza que viene siendo una tapa para asegurar la tarjeta y aislar los componentes del exterior.



(a) Pieza 1: Caja (b) Pieza 2: Tapa

Figura 4.18: Diseño 3D de contenedor para tarjeta electrónica

La figura 4.19 muestra las piezas del contenedor impresas. Tras la impresión las 2 piezas contienen todo el detalle necesario para albergar la tarjeta electrónica.



(a) Pieza 1: Caja (b) Pieza 2: Tapa

Figura 4.19: Contenedor para tarjeta electrónica elaborado

Al disponer tanto de la tarjeta electrónica como de su contenedor, se coloca ésta en el interior del contenedor. La figura 4.20 muestra el contenedor con la tarjeta en su interior desde distintos ángulos. Como se puede ver, la tarjeta queda compacta en el interior del contenedor y es visible tanto el conector de la batería como el interruptor para el funcionamiento del dispositivo. Adicionalmente, existe un agujero en la parte frontal que esta destinado para temas de soporte, ya que permite la conexión a través de un Universal Serial Bus (USB) micro a la placa D1 Mini ESP8266.

(a) Vista Superior (b) Vista frontal (c) Vista Lateral

Figura 4.20: Contenedor con tarjeta electrónica incorporada

4.4.3. Artefacto ajustable al cuerpo del paciente

La tarjeta electrónica y su contenedor constituyen el dispositivo que permite capturar y acondicionar los datos. Se requiere que este dispositivo se pueda colocar en el cuerpo de individuos, de tal manera que se pueda analizar su movilidad. Con esta consideración, se diseña y construye un cinturón que va colocado a la altura del torso. Este cinturón consta de una pieza delantera con dos bolsillos en donde se coloca la batería y el dispositivo elaborado. Otra pieza constituye en gran medida el cinturón como tal, esta pieza ocupa la parte de la espalda del paciente y se conecta a la pieza

ya mencionada a través de ganchos y correas, estas correas son ajustables. La figura 4.21 muestra las piezas que conforman el artefacto.

(a) Bolsillos de batería y contenedor (b) Parte trasera del cinturón

Figura 4.21: Cinturón ajustable para pacientes

5. Resultados y discusión

Este capítulo contiene los resultados obtenidos tras la implementación y pruebas de funcionamiento del sistema de monitorización.

5.1. Resultados

5.1.1. Dispositivo tecnológico implementado

El dispositivo tecnológico implementado consta de un prototipo funcional que identifica la posición en que se encuentra un paciente a lo largo del tiempo, el mismo se indica en la figura 5.1. El prototipo está formado por el cinturón que se coloca al paciente para la recolección de datos a través del dispositivo electrónico elaborado que lleva incluido.

Figura 5.1: Prototipo implementado para la recolección de datos

La figura 5.2 muestra un sujeto que lleva puesto el prototipo. El mismo se ajusta bien al cuerpo del individuo con las correas que tiene, esto permite que personas de distinta contextura pueda usar el prototipo.

(a) Vista frontal (b) Vista trasera

Figura 5.2: Prototipo resultante ubicado en el torso de una persona

Además, como parte del dispositivo tecnológico está la Raspberry Pi que funciona como unidad de procesamiento, aquí se lleva a cabo la clasificación de las posturas y se levanta el servicio de monitorización. En la figura 5.3 se presenta la Raspberry Pi encapsulada en un case que permite un mejor manejo del dispositivo.

Figura 5.3: Raspberry Pi utilizada

5.1.2. Pruebas de funcionamiento

Con el sistema implementado se realizan las respectivas pruebas de funcionamiento, esto tanto en un ambiente con personas sanas, así como con pacientes hospitalizados. Antes de realizar las pruebas se informa a cada voluntario a través de un consentimiento

informado, mismo que debe ser aprobado con una firma. Todo el proceso es supervisado por personal de enfermería del hospital José Félix Valdivieso del cantón Santa Isabel.

Para llevar a cabo las pruebas, en primer lugar se coloca el dispositivo en el torso del paciente, luego se enciende y se espera un tiempo mientras el dispositivo completa el proceso de calibración, con esto cumplido, los datos empiezan a ser enviados. El voluntario permanece en cada posición durante 10 minutos aproximadamente. Las pruebas se realizan con 5 sujetos, dos de ellos son pacientes hospitalizados y los demás son sujetos sanos.

Las figuras 5.4 - 5.8 muestran las posiciones que adoptaron los sujetos durante las pruebas. La información obtenida es almacenada en la base de datos para su posterior procesamiento y generación de resultados.

(a) Decúbito Supino (b) Decúbito lateral izquierdo (c) Decúbito lateral derecho

(d) Sedestación (e) Semisedestación

Figura 5.4: Pruebas realizadas con paciente A

(a) Decúbito Supino

(b) Decúbito lateral izquierdo

(c) Decúbito lateral derecho

(d) Sedestación

(e) Semisedestación

Figura 5.5: Pruebas realizadas con paciente B

(a) Decúbito Supino

(b) Decúbito lateral izquierdo

(c) Decúbito lateral derecho

(d) Sedestación

(e) Semisedestación

Figura 5.6: Pruebas realizadas con sujeto sano A

(a) Decúbito Supino (b) Decúbito lateral izquierdo (c) Decúbito lateral derecho

(d) Sedestación (e) Semisedestación

Figura 5.7: Pruebas realizadas con sujeto sano B

(a) Decúbito Supino (b) Decúbito lateral izquierdo (c) Decúbito lateral derecho

(d) Sedestación (e) Semisedestación

Figura 5.8: Pruebas realizadas con sujeto sano C

5.1.3. Validación del algoritmo de clasificación

Los resultados obtenidos en torno al rendimiento del algoritmo de clasificación k-NN son presentados en forma de gráficas de dispersión (diagrama de puntos). En los

ejes horizontal y vertical tenemos los valores del ángulo roll y pitch respectivamente de un dato clasificado, estas dos componentes conforman cada punto. Cada gráfica contiene 5 grupos de puntos identificados con distintos colores, cada uno de estos grupos corresponde a una clase o posición especifica, tal como se indica a continuación:

- Puntos verdes: Decúbito Supino
- Puntos rojos: Decúbito lateral izquierdo
- Puntos azules: Decúbito lateral derecho
- Puntos amarillos: Sedestación
- Puntos negros: Semisedestación

Cada grupo de puntos se encuentra separado de los demás debido a los valores que toman los ángulos, esto permite tener zonas de aglomeración de puntos claramente diferenciadas por posición. El punto generado para un dato correctamente clasificado se va sumando a la zona de influencia del grupo de color que le corresponde, y va agrandando la zona del mismo color. Un dato clasificado incorrectamente aparece como un punto con color distinto en la zona de un grupo de puntos que tienen el mismo color.

La figura 5.9 presenta la gráfica de puntos del paciente A, como se aprecia en este caso la clasificación es completamente correcta. Sin embargo existe algo de dispersión de los puntos para las posiciones de decúbito lateral, sobretodo del lado derecho, esto puede deberse a que el paciente se movió mas de la cuenta mientras se llevó la prueba en esta posición, sin embargo como se visualiza los datos son correctamente clasificados.

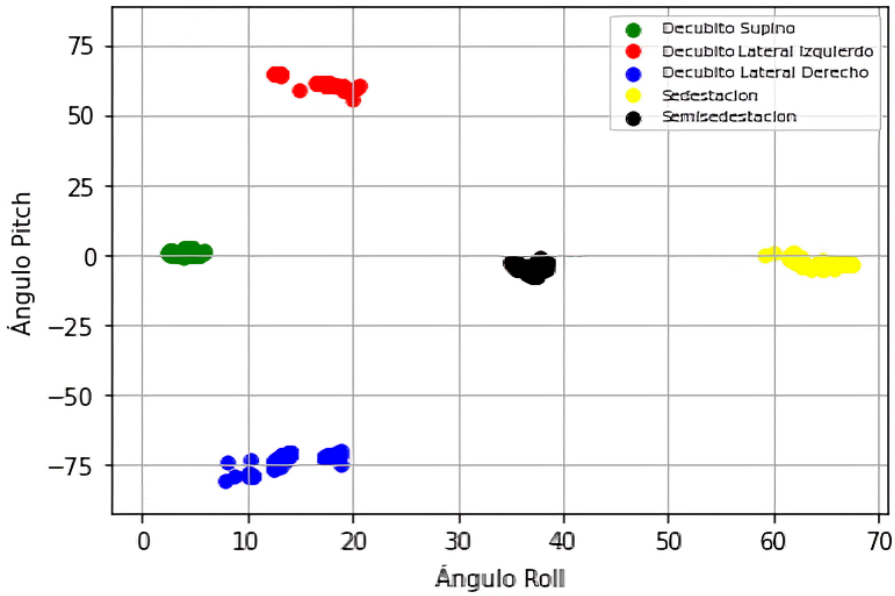


Figura 5.9: Resultados del clasificador para paciente A

Por otra parte, la figura 5.10 muestra el resultado del clasificador para el paciente B, en este caso de igual manera todos los datos son clasificados correctamente. Adicionalmente todos los puntos de cada grupo están más cerca entre sí, lo que indica que el paciente estuvo con menor movimiento durante la prueba.

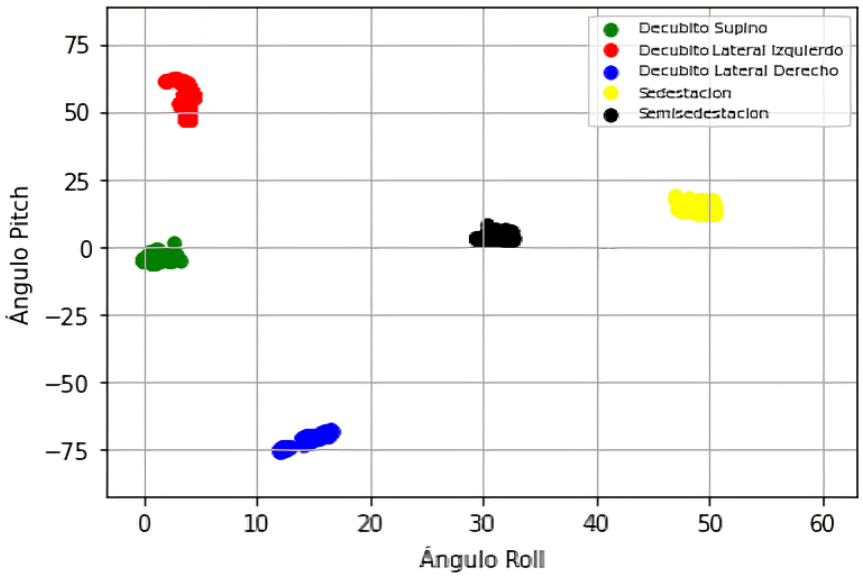


Figura 5.10: Resultados del clasificador para paciente B

Las gráficas resultantes de los sujetos sanos se presentan a continuación, en la figura 5.11 se aprecia que el clasificador funcionó correctamente en esta prueba. Además, los puntos de un mismo grupo están muy cercanos entre si y en las posiciones de decúbito lateral existe un pequeño grado de variación del ángulo roll sobretodo.

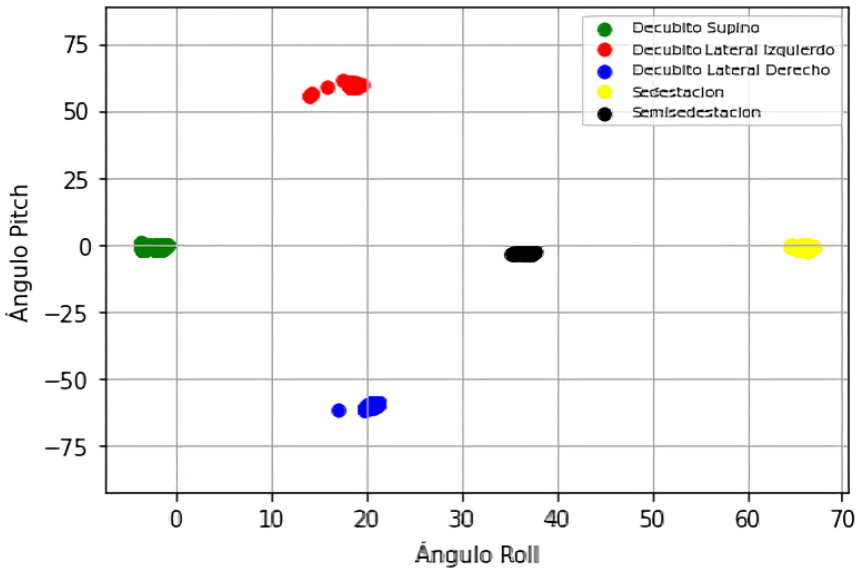


Figura 5.11: Resultados del clasificador para sujeto sano A

De manera similar, en la figura 5.12 tenemos el resultado de clasificación para el sujeto sano B, en este caso de igual manera no se aprecian errores de clasificación. Y como se ve los grupos de puntos de cada posición están bastante distantes entre si, lo que indica que el sujeto se posicionó correctamente.

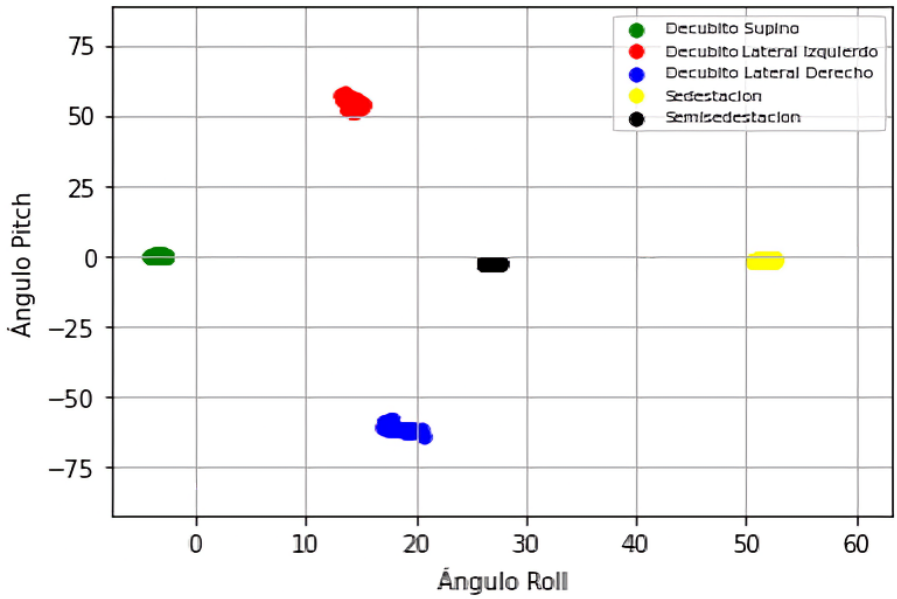


Figura 5.12: Resultados del clasificador para sujeto sano B

Finalmente, para el sujeto sano C se obtiene la gráfica de la figura 5.13 en donde de manera similar a los casos anteriores se visualiza que todos los datos fueron clasificados correctamente.

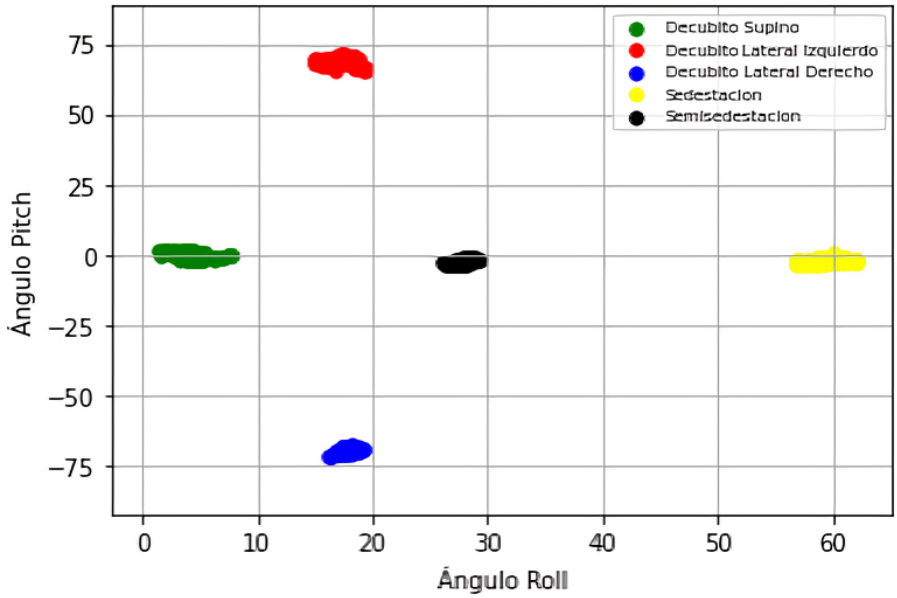


Figura 5.13: Resultados del clasificador para sujeto sano C

5.1.4. Aplicación web para monitoreo de pacientes

Con la aplicación web desplegada, es posible conectarse desde dispositivos cliente. Al ingresar se despliega la pantalla principal del sistema como aparece en la figura 5.14, en esta ventana se eligen las herramientas de la aplicación a usar.

Figura 5.14: Ventana de inicio de la aplicación web

Una de las herramientas es el test de Braden, este consta del cuestionario indicado en la figura 5.15. El cuestionario esta disponible para que el personal médico realice una evaluación rápida conforme al estado del paciente y determine a través de este test el riesgo que tiene el paciente de contraer una UPP.

Figura 5.15: Ventana con test de Braden

La otra herramienta se utiliza para el monitoreo de la posición del paciente, esta ventana presenta información como el tiempo transcurrido desde que arranca el monitoreo, el porcentaje de batería restante del prototipo y principalmente la posición del paciente cada momento, la figura 5.16 muestra esta ventana.

Figura 5.16: Ventana para monitoreo de pacientes

5.1.5. Envío de alertas

El sistema está configurado para difundir una alerta cuando el paciente se mantiene en la misma posición durante más de dos horas, este tiempo esta contemplado de acuerdo a lo indicado en trabajos como [2], sin embargo, este valor puede modificarse de acuerdo con la perspectiva médica y la necesidad del paciente. La figura 5.17, muestra el aviso enviado a través de telegram cuando se excede el tiempo programado.

Figura 5.17: Alerta enviada por exceso de tiempo en una misma posición

Otra alerta está relacionada con el porcentaje restante de batería del prototipo. Están configurados rangos de voltaje que corresponden a distintos porcentajes de batería restante. Una vez el voltaje de la batería alcanza un porcentaje menor al 10 % se envía una alerta desde la aplicación. Una captura de esta alerta se presenta en la figura 5.18. Cabe recalcar que en la notificación enviada no se incluye el valor porcentual de batería restante, sino únicamente un mensaje que indica que valor esta por debajo de un umbral.

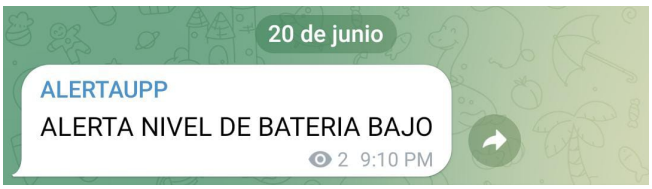


Figura 5.18: Alerta enviada por nivel bajo de batería

5.1.6. Consumo de corriente del prototipo

El prototipo tiene un consumo de corriente variable a lo largo del tiempo. En la figura 5.19 se presenta una gráfica de consumo de corriente en el transcurso de un minuto. Como se puede ver los valores varían entre 50 y 80 mA aproximadamente. El consumo promedio es de 64.03 mA.

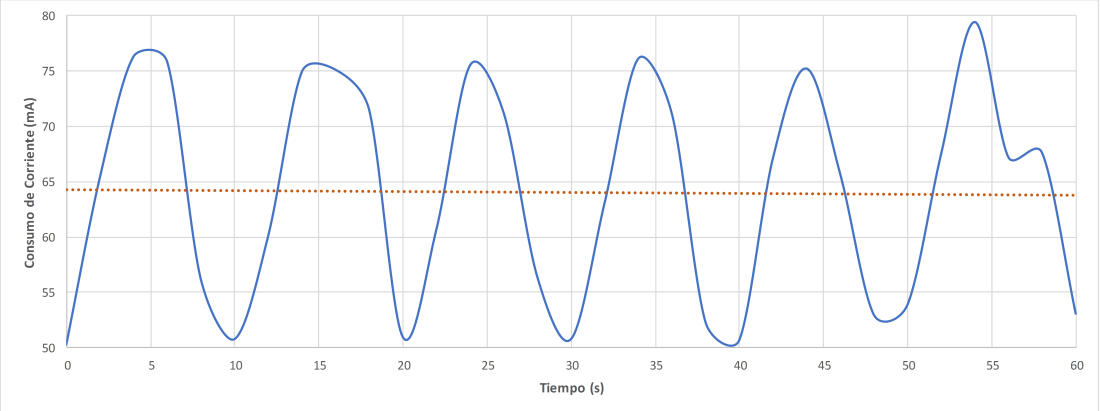


Figura 5.19: Gráfica de consumo de corriente vs tiempo

5.2. Discusión

Como parte del dispositivo tecnológico obtenido está la Raspberry Pi empleada como servidor, que en posibles escenarios de implementación donde se cuente con pocos recursos, resulta una gran opción.

Por otra parte, en torno a la evaluación del algoritmo de clasificación, todos los resultados evidencian que los conjuntos de puntos siempre están cerca entre sí, creando cinco zonas claramente marcadas que corresponden a cada posición registrada. Uno de los resultados que contiene una nube de puntos más dispersa corresponde al paciente A, una posible causa es que ésta persona es de la tercera edad y su respiración resulta más fuerte que el resto. Por otra parte, en los cinco casos no existen errores en la clasificación de posturas lo que indica que el clasificador implementado funciona bastante bien, sin embargo, el dispositivo aún puede ser puesto a prueba durante periodos mas prolongados de tiempo a fin de analizar el comportamiento.

De acuerdo a los comentarios de los voluntarios sobre las pruebas de funcionamiento, el prototipo como tal no resulta ser incómodo para los sujetos y en todos los casos indican que recomendarían el uso del prototipo y lo volverían a usar en otras ocasiones.

6. Conclusiones y recomendaciones

Este capítulo contiene las conclusiones del proyecto así como la interpretación de los resultados, las limitaciones encontradas tras la implementación y posibles mejoras a realizarse sobre el dispositivo a futuro.

6.1. Conclusiones

El sistema de monitorización implementado funciona adecuadamente, ya que determina la posición de un paciente encamado y el tiempo que lleva en dicha posición. Además, envía notificaciones de alerta al teléfono del cuidador, ayudándole a estar atento y tener un mejor cuidado del paciente.

Por otra parte, el uso de una IMU de bajo costo resulta suficientemente adecuado para recolectar datos confiables y al emplear herramientas como el filtro complementario se reduce en gran medida el error generado tanto por la medida individual del acelerómetro como del giroscopio, obteniendo así datos más limpios. En cuanto al procesamiento de estos datos, a través del algoritmo de clasificación basado en inteligencia artificial se consigue que cada dato capturado sea mapeado a una posición especifica. Para llevar esto a cabo, la Raspberry Pi que trabaja como servidor es fundamental, puesto que soporta todo el almacenamiento y procesamiento de datos, así como la aplicación web para monitoreo.

El prototipo resultante registra y almacena de manera exitosa los datos, esto se comprueba relacionando los valores obtenidos por el dispositivo y los almacenados en la base de datos. Por otra parte, el artefacto diseñado para colocar el dispositivo al paciente cumple con las expectativas al ser bastante cómodo, ajustable y fácil de acomodar en el paciente.

La aplicación web es una herramienta útil ya que permite llevar un seguimiento del estado del paciente de manera sencilla. Las alertas enviadas al dispositivo móvil ayudan a que el personal médico no esté al lado del paciente vigilandolo todo el tiempo. Además, se evidencia un correcto funcionamiento en el envío de notificaciones cuando se produce uno de los eventos alertantes configurados.

Las pruebas de funcionamiento del dispositivo llevadas a cabo con pacientes y personas sanas indican que la implementación del sistema de monitoreo desarrollado en un ambiente real es favorable.

El funcionamiento del dispositivo se limita a la duración de la batería. En caso de requerir monitorear al paciente por lapsos de tiempo prologados es recomendable disponer de una batería adicional y hacer el cambio en cuanto el nivel de batería sea crítico.

En caso de una implementación real del sistema en un centro médico, es recomendable utilizar un servidor propio de la institución que sustituya a la Raspberry Pi, esto considerando que en un escenario real, el volumen de datos a almacenar y procesar sería mucho más alto, lo que demanda más recursos de cómputo y almacenamiento.

6.2. Trabajos futuros

El dispositivo desarrollado funciona únicamente en una red local, sin embargo esto se puede mejorar para que se pueda acceder a la aplicación web a través de Internet.

Realizar mejoras sobre el prototipo, como reducir el tamaño del dispositivo empleando componentes electrónicos mas pequeños, esto a fin de generar mayor comodidad para su portador. Un segundo aspecto es el diseño del cinturón, se podría trabajar sobre este para mejorar el material, así como el método de ajuste al cuerpo del paciente.

Evaluar la funcionalidad de otros algoritmos de inteligencia artificial para la clasificación de posturas. Así como nuevas tendencias en torno a IoT que puedan ser aplicados en el sistema implementado a fin de mejorarlo.

A. Instalación y configuración de herramientas requeridas

A.1. Instalación y configuración de Mosquitto en Raspberry Pi

Raspberry OS tiene una versión reciente de Mosquitto en su repositorio de software predeterminado, por lo que se lo instala desde allí. Antes de la instalación es necesario actualizar los repositorios, las instrucciones empleadas se indican a continuación:

```
sudo apt update && sudo apt upgrade
sudo apt install -y mosquitto mosquitto-clients
```

Adicionalmente, para que Mosquitto inicie automáticamente cuando se arranque Raspberry Pi, emplear el comando siguiente:

```
sudo systemctl enable mosquitto.service
```

Para verificar la instalación, emplear el comando `mosquitto -v`, cuyo resultado de ejecución se indica en la figura A.1. Donde se aprecia que la versión de Mosquitto instalada corresponde a la 2.0.11 o superior.

```
pi@raspberrypi:~ $ mosquitto -v
1655872450: mosquitto version 2.0.11 starting
1655872450: Using default config.
1655872450: Starting in local only mode. Connections will only be possible from clients running on this machine.
```

Figura A.1: Verificación de instalación de Mosquitto

A.1.1. Habilitación de acceso remoto

Para habilitar el acceso remoto y acceder con clientes configurados en otros dispositivos hacia el broker MQTT, es necesario editar un archivo de configuración denominado `mosquitto.conf`. Para editar el archivo se emplea el comando:

```
vim /etc/mosquitto/mosquitto.conf
```

En el archivo se añaden las dos siguientes sentencias, "listener 1883", .ªllow_anonymous true.ªl final del archivo y se lo guarda. Finalmente se reinicia el servicio mosquitto en la Raspberry Pi con el comando siguiente:

```
sudo systemctl status mosquitto.service
```

Y se verifica el estado del servicio a través de la instrucción indicada en la figura A.2.

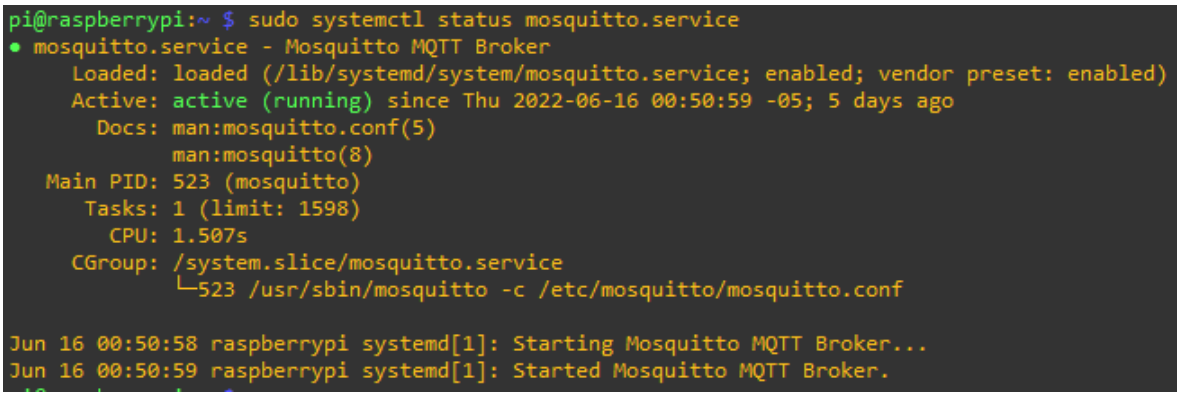


Figura A.2: Verificación del estado del servicio mosquitto

A.2. Instalación y configuración de LAMP en Raspberry Pi

Antes de comenzar el procedimiento de instalación, se ejecutan los siguientes comandos para actualizar la Raspberry Pi:

```
sudo apt update
sudo apt upgrade
```

A.2.1. Instalación de Apache2

Para instalar Apache2 en la Raspberry Pi, es necesario ejecutar el siguiente comando:

```
sudo apt install apache2 -y
```

A través de la instrucción indicada en la figura A.3, se verifica el adecuado funcionamiento del servicio apache, el mismo aparece como activo y sin errores. Otra manera de verificar el funcionamiento es acceder a través del navegador a la IP del dispositivo y se carga la web por defecto de apache.

```
pi@raspberrypi:/var/www/html/TESIS $ systemctl status apache2.service
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2022-06-16 00:51:08 -05; 5 days ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 2154 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
   Main PID: 635 (apache2)
      Tasks: 6 (limit: 1598)
        CPU: 2.827s
     CGroup: /system.slice/apache2.service
             ├─ 635 /usr/sbin/apache2 -k start
             ├─2159 /usr/sbin/apache2 -k start
             ├─2160 /usr/sbin/apache2 -k start
             ├─2161 /usr/sbin/apache2 -k start
             ├─2162 /usr/sbin/apache2 -k start
             └─2163 /usr/sbin/apache2 -k start

Jun 16 00:50:58 raspberrypi systemd[1]: Starting The Apache HTTP Server...
```

Figura A.3: Verificación del estado del servicio apache2

A.2.2. Instalación de PHP

Se instala PHP para el desarrollo de la aplicación web dinámica, a través del comando:

```
sudo apt install php -y
```

Los archivos php generados, se almacenan en el directorio /var/www/html y para probarlos se reinicia el servicio apache, como se indica a continuación:

```
sudo service apache2 restart
```

A.2.3. Instalación de MySQL (MariaDB Server)

Para instalar MySQL Server (MariaDB Server) y PHP-MySQL se emplean las siguientes instrucciones:

```
sudo apt install mariadb-server php-mysql -y
sudo service apache2 restart
```

Tras la instalación, es recomendable ejecutar la instrucción indicada a continuación, para asegurar la instalación de MySQL:

```
sudo mysql_secure_installation
```

Con esto, el formulario indicado en la figura A.4 aparece en la terminal, el mismo se llena de la siguiente manera:

- Ingreso de la contraseña actual para root.

- Escribir Y y presionar Entrar para establecer la contraseña de root.
- Escribir una contraseña cuando se solicite, Nueva contraseña.
- Escribir Y para eliminar usuarios anónimos.
- Escribir Y para no permitir el inicio de sesión raíz de forma remota.
- Escribir Y para eliminar la base de datos de prueba y el acceso a ésta.
- Escribir Y para recargar las tablas de privilegios ahora.

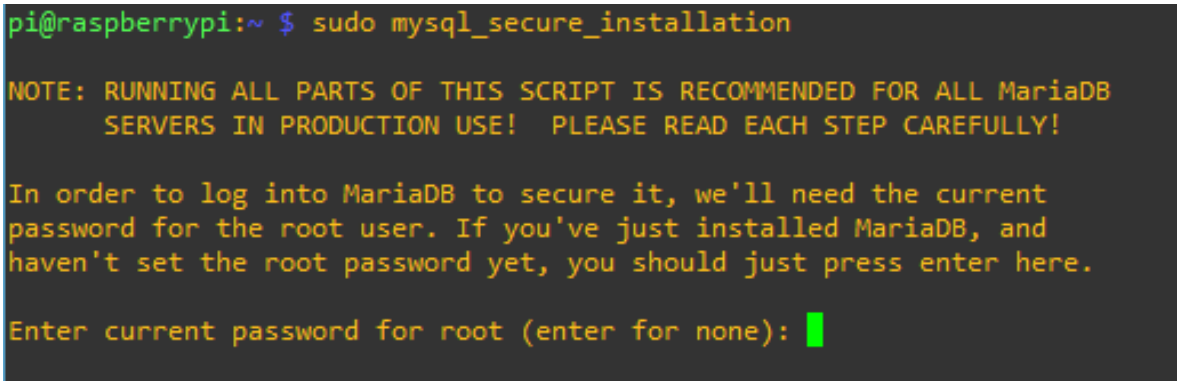


Figura A.4: Asegurando la instalación de MySQL

A.2.4. Instalación de phpMyAdmin

phpMyAdmin es la herramienta destinada a manejar la administración de MySQL usando una interfaz web. Para instalar phpMyAdmin en la Raspberry Pi, se emplea la siguiente instrucción:

```
sudo apt install phpmyadmin -y
```

Se presenta la ventana de instalación de phpmyadmin, tal cual como se indica en la figura A.5.

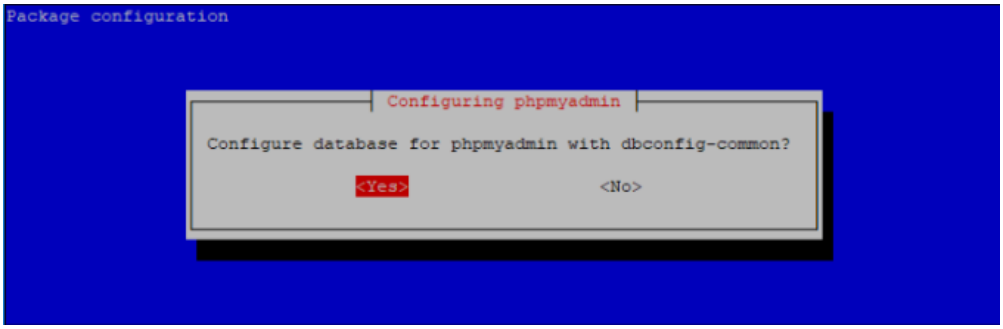


Figura A.5: Ventana de instalación de phpmyadmin

Aquí se selecciona Yes, para configurar phpmyadmin con dbconfig-common. Y se establecen las siguientes opciones durante la configuración:

- Servidor Web, seleccionar Apache2.
- Configurando phpmyadmin, selecionar OK.
- Configurar la base de datos para phpmyadmin con dbconfig-common, seleccionar Yes.
- Escribir una contraseña y presionar OK.

Luego, se ejecutan las siguientes instrucciones para habilitar la extensión PHP MySQL y reiniciar el servicio de apache2 para que los cambios tengan efecto.

```
sudo phpenmod mysqli
sudo service apache2 restart
```

Finalmente, para evitar problemas al ingresar a la interfaz web de pypmyadmin, se mueve el directorio "phpmyadmin.ª la ruta /var/www/html, a traves de la siguiente instrucción:

```
/var/www/html $ sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```

A.3. Implementación de alerta en Telegram

A.3.1. Creación de bot

En un chat propio de la aplicación denominado BotFather, ingresamos el mensaje `/newbot`, esto despliega un mensaje indicándonos que enviemos el nombre del nuevo Bot y un nombre de usuario. El programa realiza validaciones y pedirá ingresar un nuevo nombre si hemos colocado uno ya registrado. En este caso el nombre del bot es `ALERTA` y el nombre de usuario `/AlertaTesisbot`, como indica la figura A.6.

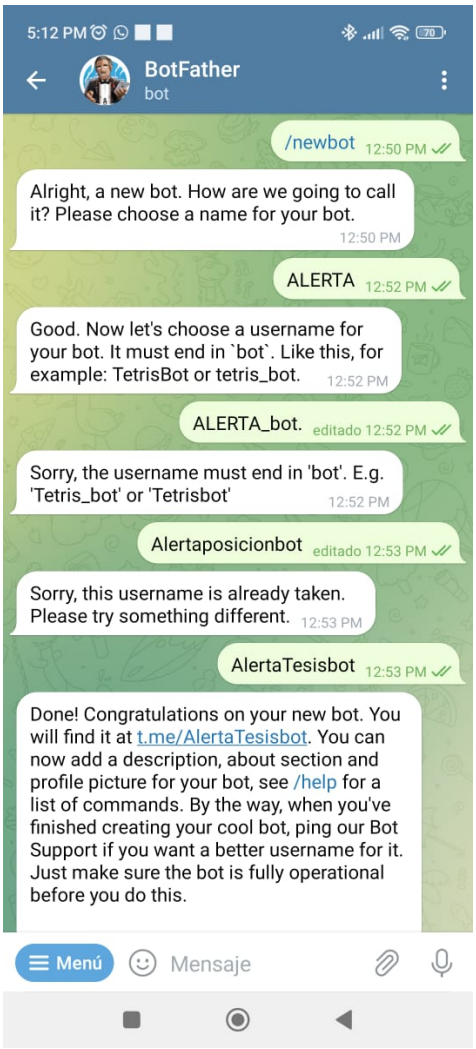


Figura A.6: Creación de bot para alertas de telegram

A.3.2. Creación de canal de difusión

El bot creado previamente envía los mensajes de alerta a través de un canal de difusión. Para crear el canal indicado, hacer clic en el ícono azul que contiene un lápiz, esto despliega una nueva ventana donde seleccionamos `Nuevo canal`. La interfaz para iniciar con la configuración es similar a lo presentado en la figura A.7.

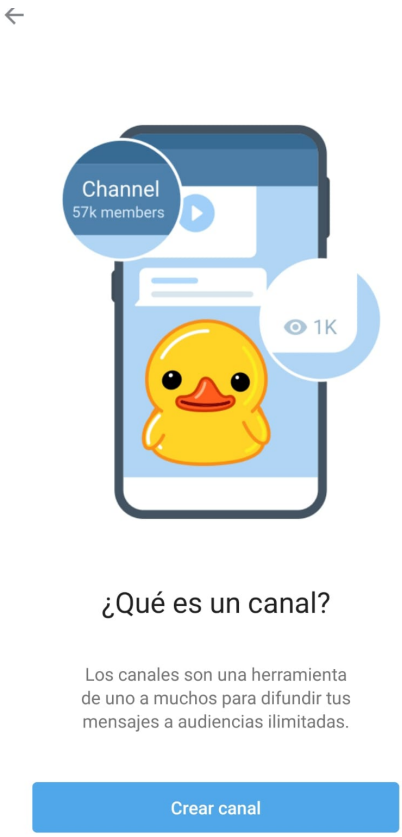


Figura A.7: Interfaz inicial para creación de canal de telegram

Es necesario ingresar un nuevo nombre del canal, y opcional una descripción del mismo en una ventana similar presentada en la figura A.8.

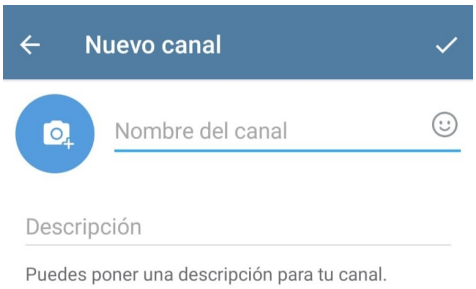


Figura A.8: Campos a llenar para crear canal de difusión

Finalmente, se escoge el tipo de canal. Para nuestro propósito seleccionamos `Canal privado`. Esto para que el canal no esté disponible en la búsqueda de la aplicación y que se puede

ingresar con aprobación del administrador.

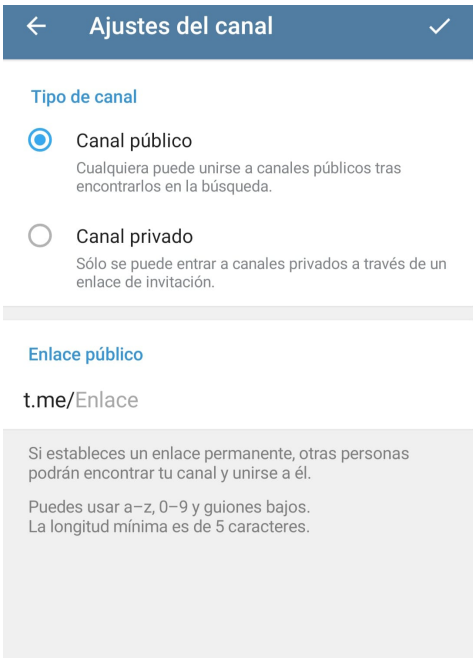


Figura A.9: Selección de tipo de canal a crear

B. Características de dispositivos

B.1. Características ESP8266

- Voltaje de Alimentación: 5 VDC.
- Voltaje de Entradas/Salidas: 3.3 VDC.
- CPU: Tensilica Xtensa LX3 (32 bit).
- Frecuencia de Reloj: 80MHz/160MHz.
- Instruction RAM: 32KB.
- Data RAM: 96KB.
- Memoria Flash Externa: 4 MB.
- Pines Digitales GPIO: 11 (3.3 V).
- Pin Analógico ADC: 1 (0-1 V).
- Puerto serial UART: 1 (3.3 V).
- Corriente Standby: 40 uA.
- Corriente Pico: 400 mA.
- Consumo corriente promedio: 70 mA.
- Conectividad 802.11 b/g/n.
- Wi-Fi Direct (P2P), soft-AP.
- Stack de Protocolo TCP/IP integrado.
- PLLs, reguladores, DCXO y manejo de poder integrados.
- Potencia de salida de +19.5 dBm en modo 802.11b.

B.2. Características Raspberry Pi 3B+

La Raspberry Pi 3 Model B+ es la revisión final de la gama Raspberry Pi 3. [75]:

- SoC Broadcom BCM2837B0, Cortex-A53 (ARMv8) de 64 bits a 1,4 GHz.
- SDRAM LPDDR2 de 1 GB.

- LAN inalámbrica de 2,4 GHz y 5 GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE.
- Gigabit Ethernet sobre USB 2.0 (rendimiento máximo de 300 Mbps).
- Encabezado GPIO de 40 pines extendido.
- HDMI de tamaño completo.
- 4 puertos USB 2.0.
- Puerto de cámara CSI para conectar una cámara Raspberry Pi.
- Puerto de pantalla DSI para conectar una pantalla táctil Raspberry Pi.
- Salida estéreo de 4 polos y puerto de video compuesto.
- Puerto Micro SD para cargar su sistema operativo y almacenar datos.
- Entrada de alimentación de CC de 5 V/2,5 A.
- Compatibilidad con alimentación a través de Ethernet (PoE) .

C. Descripción de Scripts desarrollados

C.1. Script de Arduino IDE para el módulo D1 MINI ESP8266

El entorno de desarrollo integrado de Arduino es la aplicación utilizada para editar, depurar y grabar el programa desarrollado sobre la placa D1 MINI ESP8266.

C.1.1. Librerías agregadas y variables

Para el funcionamiento adecuado del programa desarrollado es importante cargar algunas librerías disponibles para el control y calibración de la IMU MPU6050 (`I2Cdev.h`, `MPU6050.h`, `Wire.h`), para la configuración del cliente MQTT y la conexión del módulo a la red Wifi (`PubSubClient.h`, `ESP8266WiFi.h`).

```
// Librerias para el control y calibración de la IMU MPU6050
// La libreria MPU6050.h requiere I2Cdev.h, I2Cdev.h requier Wire.h
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

// Libreria para MQTT
#include "PubSubClient.h"
// Libreria para conexión del ESP8266 a la red WiFi
#include "ESP8266WiFi.h"
```

Las variables definidas son utilizadas a lo largo del script dentro de las distintas secciones que lo componen, esto incluye la conexión MQTT, la calibración de la IMU MPU6050, el cálculo de ángulos y la definición del indicador de carga.

```
// ***************+ Variables para conexión MQTT +*****************
char msg[25];
int count = 0;
String clientId;
const int mqtt_port = 1883;
const char *mqtt_user = "";
const char *mqtt_pass = "";
const char *mqtt_server = "192.168.2.102";
const char *root_topic_publish = "/tesis/pruebas/out";
```

```
const char *root_topic_subscribe = "/tesis/pruebas/in";
// Asignación de cliente mqtt (client)
PubSubClient client(espClient);


//************* Variables para calibración de MPU6050 *************
// La dirección del MPU6050 puede ser 0x68 o 0x69, dependiendo
// del estado de AD0. Si no se especifica, 0x68 estará implicito
MPU6050 sensor;
int counter = 0;
//Valor de los offsets
int ax_o, ay_o, az_o;
int gx_o, gy_o, gz_o;
//Variables usadas por el filtro pasa bajos
long f_ax, f_ay, f_az;
int p_ax, p_ay, p_az;
long f_gx, f_gy, f_gz;
int p_gx, p_gy, p_gz;


// *********+ Variables para el calculo de ángulos +**********
float dt;
String data;
String values;
long time_prev;
float accel_ang_x;
float accel_ang_y;
float giros_ang_x;
float giros_ang_y;
float giros_ang_z;
int16_t ax, ay, az;
int16_t gx, gy, gz;
float ang_x, ang_y;
float ang_x_prev, ang_y_prev, ang_z_prev;


// ************+ Variables para indicador de carga +************
float volt;
int sensorValue = 0;
const int LED1 = D5;
const int LED2 = D6;
const int analogInPin = A0;
```

C.1.2. Conexión a la red WiFi, funcion setup_wifi()

Para conectar la placa D1 MINI ESP8266 a una red WiFi se implementa la función indicada a continuación, ésta contiene las instrucciones para establecer un modo de conexión, cargar los parámetros de red y establecer la conexión. Además, contiene un bucle que persiste hasta que la placa busque y logre conectarse.

```
void setup_wifi() {
  delay(10);                              // Retraso inicial de 10 ms
  //Se establece el modo de conexion (Estación), se cargan los parámetros
  // de la red y se inicia la conexión del modulo ESP8266 a la red
  WiFi.mode(WIFI_STA);
  WiFi.config(ip, gateway, subnet);
  WiFi.begin(ssid, password);
  Serial.println();
  Serial.print("Conectando a ssid: ");
  Serial.println(ssid);
  // Bucle de espera mientras el módulo se conecta a la red
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Impresión de mensaje de conexión exitosa y dirección IP asignada
  Serial.println("");
  Serial.println("Conectado a red WiFi!");
  Serial.println("Dirección IP: ");
  Serial.println(WiFi.localIP());
  delay(2000);
}
```

C.1.3. Conexión al broker MQTT, funcion reconnect()

Esta función se ejecuta mientras el cliente MQTT no esté conectado al broker, internamente crea un ID de cliente para la placa y trata de conectarse al broker con las credenciales definidas en las variables. En caso de no lograr la conexión hace un reintento luego de cinco segundos. El código implementado para esta función es el siguiente:

```
//******************* Conexión MQTT ******************
void reconnect() {
  // Bucle que se ejecuta mientras el cliente mqtt no este conectado
  while (!client.connected()) {
    Serial.print("Intentando conexión Mqtt...");
    // Creación de un client ID
    clientId = "IOTICOS_H_W_";
```

```
    clientId += String(random(0xffff), HEX);
    // Intento de conexión al broker
    if (client.connect(clientId.c_str(), mqtt_user, mqtt_pass)) {
      Serial.println("Conectado!");
    } else {
      Serial.print("falló :( con error -> ");
      Serial.print(client.state());
      Serial.println("Reintentando en 5 segundos");
      delay(5000);
    }
  }
}
```

C.1.4. Inicialización de funciones de ejecución única

En esta sección se setean algunas funciones que lleva a cabo el microcontrolador y ejecuta únicamente una vez, como son, apertura del puerto serial a 115200 baudios, llamada a la función para la conexión WiFi, inicialización de la conexión MQTT con el broker y de la IMU con la placa a través de la comunicación I2C. Además, se establecen 2 puertos digitales de la placa como salidas.

```
void setup() {
  Serial.begin(115200);                    // Iniciando puerto serial
  setup_wifi();                            // Conexion WiFi del módulo
   ESP8266
  client.setServer(mqtt_server, mqtt_port); // Abriendo conexión con el
   broker mqtt
  Wire.begin();                            // Iniciando I2C
  sensor.initialize();                     // Iniciando el sensor
   MPU6050
  // Prueba de conexión del sensor MPU6050
  if (sensor.testConnection())
  Serial.println("Sensor iniciado correctamente");
  else Serial.println("Error al iniciar el sensor");
  // Se establecen los pines de LEDs como salidas
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
```

C.1.4.1. Calibración de la IMU

Esta sección está modificando constantemente los offset en busca de eliminar el error y obtener el valor deseado. Para ello, es necesario leer los offsets iniciales del sensor, así como los valores del acelerómetro y giroscopio, estos valores se estabilizan con un

filtro y cada 50 lecturas se hace un ajuste en los offsets. Esto busca que las lecturas filtradas tiendan a:

- Aceleración: $p\_ax = 0$, $p\_ay = 0$, $p\_az = +16384$.
- Velocidad angular: $p\_gx = 0$, $p\_gy = 0$, $p\_gz = 0$.

Esta sección corre bajo un bucle que persiste hasta que los valores de los offsets se acerquen lo suficiente a los valores indicados. A continuación, el detalle del código implementado:

```
//***************+ Calibración del sensor MPU6050 +****************
// Lectura de los offsets iniciales del sensor
ax_o = sensor.getXAccelOffset();
ay_o = sensor.getYAccelOffset();
az_o = sensor.getZAccelOffset();
gx_o = sensor.getXGyroOffset();
gy_o = sensor.getYGyroOffset();
gz_o = sensor.getZGyroOffset();
Serial.println("Calibrando, no mover IMU");

// Se ejecuta el bucle hasta que finalice el ajuste de offsets
while (p_ax > 0 || p_ay < 0 || p_az < 16384) {

  // Lectura de aceleraciones y velocidades angulares del sensor
  sensor.getAcceleration(&ax, &ay, &az);
  sensor.getRotation(&gx, &gy, &gz);

  // Filtrado de de lecturas de aceleraciones
  f_ax = f_ax - (f_ax >> 5) + ax;
  p_ax = f_ax >> 5;
  f_ay = f_ay - (f_ay >> 5) + ay;
  p_ay = f_ay >> 5;
  f_az = f_az - (f_az >> 5) + az;
  p_az = f_az >> 5;
  // Filtrado de de lecturas de velocidades angulares
  f_gx = f_gx - (f_gx >> 3) + gx;
  p_gx = f_gx >> 3;
  f_gy = f_gy - (f_gy >> 3) + gy;
  p_gy = f_gy >> 3;
  f_gz = f_gz - (f_gz >> 3) + gz;
  p_gz = f_gz >> 3;

  //Cada 50 lecturas corregir el offset
  if (counter == 50) {
    //Impresión de offsets filtrados a través del monitor serie
    Serial.print("promedio:"); Serial.print("\t");
    Serial.print(p_ax); Serial.print("\t");
```

```
      Serial.print(p_ay); Serial.print("\t");
      Serial.print(p_az); Serial.print("\t");
      Serial.print(p_gx); Serial.print("\t");
      Serial.print(p_gy); Serial.print("\t");
      Serial.println(p_gz);

      //Calibración del acelerometro a 1g en el eje z (ajuste del offset)
      if (p_ax > 0) ax_o--;
      else ax_o++;
      if (p_ay > 0) ay_o--;
      else ay_o++;
      if (p_az - 16384 > 0) az_o--;
      else az_o++;
      // Se setean los nuevos offsets del acelerómetro
      sensor.setXAccelOffset(ax_o);
      sensor.setYAccelOffset(ay_o);
      sensor.setZAccelOffset(az_o);

      //Calibración el giroscopio a 0grad/s en todos los ejes (ajuste del
    offset)
      if (p_gx > 0) gx_o--;
      else gx_o++;
      if (p_gy > 0) gy_o--;
      else gy_o++;
      if (p_gz > 0) gz_o--;
      else gz_o++;
      // Se setean los nuevos offsets del giroscopio
      sensor.setXGyroOffset(gx_o);
      sensor.setYGyroOffset(gy_o);
      sensor.setZGyroOffset(gz_o);

      //Reseteo del contador incremental a cero
      counter = 0;
    }
    // Incremento en una unidad del contador incremental
    counter++;
  }
}
```

C.1.5. Ejecución de instrucciones en void_loop()

Esta seccion constituye a parte principal del script, ya que se ejecuta un número infinito de veces. Al encenderse el equipo se ejecuta el código del setup y luego se entra al loop, el cual se repite de forma indefinida hasta que se apague o se reinicie la placa. Dentro de esta sección, lo primero a realizar es la conexión de la placa al broker MQTT,

para eso se llama a la función `reconnect()`, como se indica a continuación:

```
void loop() {
  // Comprobación del cliente conectado al broker mqtt
  if (!client.connected()) {
    reconnect();
  }
```

C.1.5.1. Cálculo de ángulos de inclinación

Para calcular el ángulo de inclinación en cada eje, se leen las aceleraciones y velocidades angulares, a partir de estas lecturas se determina el ángulo de inclinación de manera individual, a traves de operaciones establecidas, como se indica a continuación:

```
  //**************+ Cálculo de ángulos de inclinación +***************

  // Lectura de aceleraciones y velocidades angulares del sensor
  sensor.getAcceleration(&ax, &ay, &az);
  sensor.getRotation(&gx, &gy, &gz);
  // Intervalo de tiempo para el cálculo de angulo con giroscopio
  dt = (millis() - time_prev) / 1000.0;
  //Tiempo de ejecucion del programa
  time_prev = millis();
  // Cálculo de ángulos de inclinación con acelerómetro
  accel_ang_x = atan(ay / sqrt(pow(ax, 2) + pow(az, 2))) * (180.0 / 3.14)
   ;
  accel_ang_y = atan(-ax / sqrt(pow(ay, 2) + pow(az, 2))) * (180.0 /
   3.14);

  //Cálculo de ángulos de rotación con giroscopio
  giros_ang_x = ang_x_prev + (gx / 131) * dt;
  giros_ang_y = ang_y_prev + (gy / 131) * dt;
  giros_ang_z = ang_z_prev + (gz / 131) * dt;
```

C.1.5.2. Implementación de filtro complementario

Posteriormente, se emplea el filtro complementario para obtener una medición más precisa del ángulo en cada eje y eliminar los errores asociados a las medidas individuales del acelerómetro y giroscopio. El procedimiento es el siguiente:

```
  //Integración de filtro complementario para el cálculo de ángulos
  ang_x = 0.98 * giros_ang_x + 0.02 * accel_ang_x;
  ang_y = 0.98 * giros_ang_y + 0.02 * accel_ang_y;

  // Actualización de los ángulos previos al final de la iteración
  ang_x_prev = ang_x;
```

```
ang_y_prev = ang_y;
ang_z_prev = giros_ang_z;
```

C.1.6. Implementación de indicador de carga de batería

Aquí se leen los valores que ingresan a través del puerto analógico A0 de la placa D1 MINI ESP8266, convierte los valores a una medida real de nivel de voltaje restante en la batería, al tomar como referencia que 4.11 V es el valor máximo. Luego, a través de los dos puertos digitales previamente definidos, la placa enciende un indicador que cambia de color a rojo cuando el nivel de batería esta demasiado bajo.

```
//*********** Lectura de voltaje como indicador de nivel de carga
 ***********

// Lectura de voltaje de batería por el pin A0 del módulo ESP8266
sensorValue = analogRead(analogInPin);
volt = sensorValue * 4.11 / 1023;

// Condiciones para ON/OFF de LEDs indicadores de nivel de carga
if (volt > 3) {
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, LOW);
} else {
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, HIGH);
}
```

C.1.7. Conexión y envío de datos al servidor MQTT

Los valores angulares en el eje X y Y, así como el nivel de voltaje son almacenados en una variable de tipo string que compone el set de datos a publicar a través de un tema creado bajo el protocolo MQTT hacia el broker. La publicación de un nuevo dato se produce cada segundo. La sección de código implementada se indica a continuación:

```
// ************ Publicación de ángulos y voltaje ********************

// Impresión de ángulos en los 3 ejes con referencias de +/-90grad
values = "90, " + String(ang_x) + "," + String(ang_y) + "," + String(
 giros_ang_z) + ", -90";
Serial.println(values);

// Publicación de ángulos de inclinación roll(ang_x), pitch(ang_y)
// y voltaje de bateria medido, en el broker MQTT cada 2 segundos
data = String(ang_x) + "," + String(ang_y) + "," + String(volt);
```

```
  if (client.connected() && count == 100) {
    data.toCharArray(msg, 20);
    client.publish(root_topic_publish, msg);
    count = 0;
  }
  delay(10);
  count++;
}
```

C.2. Script de python para recepción de datos vía MQTT y su almacenamiento

Un script desarrollo en python cargado en la Raspberry Pi que actúa de broker y servidor de base de datos, permite capturar datos a través del protocolo MQTT y almacenarlos en una tabla en la base de datos creada.

C.2.1. Librerías requeridas importadas

Se instalan e importan las librerías indicadas a continuación para el adecuado funcionamiento del script diseñado. Estas son, `sys` que proporciona acceso a variables y funciones específicas del sistema, `paho.mqtt.client` para establecer la conexión a través del protocolo MQTT, `mysql.connector` para la gestión de bases de datos y y `datetime` para determinar la fecha y hora actual del sistema.

```
import sys
import paho.mqtt.client as mqtt
import mysql.connector
from datetime import datetime
```

C.2.2. Funciones definidas en el script

Este script se construye a partir de algunas funciones que se detallan a continuación:

C.2.2.1. Conexión a la base de datos, función ConexionDB()

Establece la conexión del equipo (Raspberry Pi) a la base de datos TESIS. El hostname del equipo que abre la conexión así como las credenciales de un usuario registrado en la base son parámetros requeridos en este procedimiento. A continuación el detalle de esta función:

```
# Funcion para conexión a la base de datos TESIS
def ConexionDB():
    conexion = mysql.connector.connect(host="localhost",
                                       user="admin",
                                       passwd="1234",
                                       database="TESIS")
    return conexion
```

C.2.2.2. Función para conexión al broker, función on_connect()

Esta función presenta un mensaje que indica una conexión satisfactoria del cliente definido localmente con el broker, además, suscribe al cliente al tema común donde esta publicando los datos la placa D1 MINI ESP8266. A continuación, el detalle de esta sección del script:

```
# Funcion que imprime el resultado del intento de conexion y realiza la
# suscripcion al topic /tesis/prueba/out
def on_connect(client, userdata, flags, rc):
    print('connected (%s)' % client._client_id)
    client.subscribe(topic='/tesis/pruebas/out', qos=1)
```

C.2.2.3. Función para almacenar dato nuevo en la base de datos, on_message()

Esta función abre una conexión con la base de datos TESIS para insertar el dato capturado por el cliente MQTT de la Raspberry, en la tabla DATOS. Los valores insertados son un ID del dato, la fecha y hora al momento y el payload que el cliente recuperó del broker, este dato es un string separado por comas que contiene el valor de los angulos roll, pitch (eje X, Y) y el nivel de voltaje de la bateria del prototipo. La siguiente sección de código muestra a detalle el contenido de esta función:

```
# Funcion para conexion y almacenamiento en base de datos TESIS
def on_message(client, userdata, message):
    # Impresión de datos recogidos del cliente MQTT
    print('------------------------------')
    print('payload: %s' % message.payload)
    # Conexion a la base de datos TESIS
    conexion1 = ConexionDB()
    # Obtención de fecha y hora actual
    timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    # Datos recogidos del cliente MQTT
    dato = message.payload
    # Armado de query SQL para insercion en la tabla DATOS
    query = """INSERT INTO `DATOS` (`ID`, `FECHA`, `MEDICION`)
                VALUES (%s, %s, %s)"""
```

```
    datos = (None, timestamp, dato)
    # Creación de cursor para ejecución de query
    cursor1 = conexion1.cursor()
    # Llamada al método execute para ejecutar la query diseñada
    cursor1.execute(query, datos)
    # Llamada al método commit para afirmar cambios en la base de datos
    conexion1.commit()
    # Cierre de conexión con la base de datos
    conexion1.close()
```

C.2.2.4. Función principal del script, main()

Esta función crea una instancia de cliente MQTT para la Raspberry Pi y realiza la conexión del cliente al broker, a través del localhost por el puerto 1883. Posterior a ello, se ejecuta la función `on_connect()` que el cliente llama al establecer una conexión con el broker. Le sigue la ejecución de la función `on_message()` que de igual manera el cliente llama al recibir un mensaje en el tema suscrito. Finalmente se establece un método para que el programa se ejecute de manera indefinida.

```
# Función principal para ejecución del programa
def main():
    # Creación de una instancia de cliente con MQTT
    client = mqtt.Client(client_id='esp8266-sub', clean_session=False)
    # Función callback que el cliente llama al conectarse al broker
    client.on_connect = on_connect
    # Función callback que el cliente llama al recibir un mensaje
    client.on_message = on_message
    # Conexión al broker MQTT en el localhost
    client.connect(host='127.0.0.1', port=1883)
    # Llamada al método para ejecutar  indefinidamente el programa
    client.loop_forever()
```

C.2.3. Ejecución de la función principal del script

Como sección principal del script, se define la condición para la ejecución de la función `main()`, lo que permite que el script arranque y se mantenga en ejecución indefinida, en caso de salir del bucle, se define una salida limpia sin errores de ejecución para terminar.

```
# Condición para ejecución de la función main
if __name__ == '__main__':
    main()

sys.exit(0)
```

C.3. Script de python para implementación del algoritmo de clasificación

C.3.1. Librerías requeridas importadas

Entre las librerías instaladas e importadas en este script están `numpy`, `matplotlib` para el manejo de vectores, matrices y gráficos respectivamente. Además, `mysql.connector` para la gestión de bases de datos, y `datetime`, `time` para el manejo de tiempos de ejecución.

```
# -*- coding: utf-8 -*-
# !/usr/bin/python

import numpy as np
import mysql.connector
from datetime import datetime
import time
```

C.3.2. Funciones definidas en el script

En el script se definen 2 funciones, la primera es `DisEuclidiana`, ésta calcula la distancia euclidiana entre dos puntos con coordenadas (X, Y). Con la segunda función se establece la conexión del equipo (Raspberry Pi) a la base de datos TESIS. A continuación el detalle de esta sección:

```
# Funcion que calcula la distancia euclidiana
def DisEuclidiana(x, y):
    return np.sqrt(np.sum((x-y)**2))


# Funcion para conexión a la base de datos TESIS
def ConexionDB():
    conexion = mysql.connector.connect(host="192.168.2.102",
                                       user="admin",
                                       passwd="1234",
                                       database="TESIS")
    return conexion
```

C.3.3. Matriz con muestras de entrenamiento

A continuación se definen las muestras de entrenamiento a ser utilizadas como referencia en la implementación del algoritmo de clasificación. Existen 10 muestras por

cada postura que siguen el siguiente orden; decúbito supino, decúbito lateral izquierdo, decúbito lateral derecho, sedestación y semi sedestación. Estas se agrupan en una matriz de nombre X que contiene todos estos valores.

```
# Muestras de entrenamiento - Decubito supino
x1 = np.array([[0], [0]])
x2 = np.array([[0.18], [0.3]])
x3 = np.array([[4.74], [0.1]])
x4 = np.array([[14.3], [1.2]])
x5 = np.array([[3.25], [1.28]])
x6 = np.array([[2.6], [1.38]])
x7 = np.array([[1.12], [2.45]])
x8 = np.array([[10.70], [3.1]])
x9 = np.array([[0.6], [1.3]])
x10 = np.array([[8.1], [2.1]])
# Muestras de entrenamiento - Decubito lateral izquierdo
x11 = np.array([[0], [90]])
x12 = np.array([[1.1], [87]])
x13 = np.array([[0.8], [88.5]])
x14 = np.array([[0.6], [78.2]])
x15 = np.array([[2], [89]])
x16 = np.array([[5.2], [80.2]])
x17 = np.array([[-2.2], [80]])
x18 = np.array([[4.1], [75.2]])
x19 = np.array([[7], [81]])
x20 = np.array([[4], [80]])
# Muestras de entrenamiento - Decubito lateral derecho
x21 = np.array([[0], [- 90]])
x22 = np.array([[-0.4], [-78.3]])
x23 = np.array([[-1], [-77]])
x24 = np.array([[2], [-78.2]])
x25 = np.array([[0], [-75]])
x26 = np.array([[0.5], [-84]])
x27 = np.array([[-4], [-85]])
x28 = np.array([[-7.3], [-79.6]])
x29 = np.array([[5.6], [-88.6]])
x30 = np.array([[0.5], [-87]])
# Muestras de entrenamiento - Semisedestación
x31 = np.array([[50], [0]])
x32 = np.array([[55.44], [2.04]])
x33 = np.array([[75.59], [1.98]])
x34 = np.array([[80.01], [1.10]])
x35 = np.array([[65.24], [0.91]])
x36 = np.array([[60.06], [0.97]])
x37 = np.array([[62.2], [7.2]])
x38 = np.array([[58], [-4.22]])
x39 = np.array([[68], [-0.21]])
x40 = np.array([[52.1], [2.33]])
```

```
# Muestras de entrenamiento - Sedestación
x41 = np.array([[25], [0.2]])
x42 = np.array([[30], [1.2]])
x43 = np.array([[35], [2.3]])
x44 = np.array([[45], [0.1]])
x45 = np.array([[42], [4.2]])
x46 = np.array([[41], [-3.6]])
x47 = np.array([[38], [0.5]])
x48 = np.array([[34.5], [6.7]])
x49 = np.array([[38.6], [-5.7]])
x50 = np.array([[39], [-2.3]])

# Matriz con muestras de entrenamiento
X = np.concatenate((x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12,
    x13,
                    x14, x15, x16, x17, x18, x19, x20, x21, x22, x23, x24
    ,
                    x25, x26, x27, x28, x29, x30, x31, x32, x33, x34, x35
    ,
                    x36, x37, x38, x39, x40, x41, x42, x43, x44, x45, x46
    ,
                    x47, x48, x49, x50), axis=1)
```

C.3.4. Variables de inicio

Establecer el valor de K correspondiente al numero de vecinos mas cercanos, en tres y contar el numero de muestras de entrenamiento, son las instrucciones ejecutadas en esta sección.

```
# Se establece el valor de K (# de vecinos mas cercanos)
K = 3
# Se identifica el número de muestras de entrenamiento
muestras = X.shape[1]
```

C.3.5. Lectura de datos de la base de datos

En el transcurso del script se ejecuta un bucle de forma indefinida, hasta enviar una señal que pare la ejecución del script. Dentro del bucle, se establece la conexión con la base de datos para extraer los datos previamente almacenados con el script anterior. Dicho dato contiene tres valores, los dos ángulos y voltaje restante de la batería. Sin embargo, en este punto únicamente interesan los datos de ángulos por lo que se extraen estos dos valores en un array.

```
# Bucle para ejecución del programa indefinidamente
while 1 == 1:
    # Conexión a la base de datos TESIS
    conexion1 = ConexionDB()
    # Query SQL para extraer valores de la tabla DATOS
    query1 = "select MEDICION from DATOS order by ID desc limit 1"
    # Creación de cursor para ejecución de query
    cursor1 = conexion1.cursor()
    # Llamada al método execute para ejecutar la query diseñada
    cursor1.execute(query1)
    # Se extrae el dato del campo consultado en la base de datos
    for fila in cursor1:
        datos_query = fila
    # Cierre de conexión con la base de datos
    conexion1.close()
    # Se genera un array al separar por comas los elementos del dato
    array_datos = datos_query[0].split(sep=',')
    print(array_datos)
```

C.3.6. Algoritmo de clasificación

En esta sección se implementa el algoritmo de clasificación K-nearest Neighbor, a través del cálculo de la distancia euclidiana entre las muestras de entrenamiento y un nuevo dato capturado. Se ordenan las distancias de menor a mayor y de esta manera se definen los K vecinos mas cercanos que corresponden a tres valores.

```
    # Se arma un array con las posiciones 0 y 1 del vector de datos,
    # estos valores corresponden a los ángulos roll y pitch medidos
    Y = np.array([[float(array_datos[0])], [float(array_datos[1])]])
    # Se establece un array con tamaño igual al # de grupos a clasificar
    clases = np.zeros(5)
    # Array no inicializado con tamaño igual al # de muestras iniciales
    distancias = np.empty(muestras)
    # Cálculo de la distancia entre el dato medido y cada muestra inicial
    for n in range(muestras):
        distancias[n] = DisEuclidiana(X[:, n], Y[:, 0])
    # Array con indices del array "distancias" ordenado de menor a mayor
    valor
    distancias_ord = np.argsort(distancias)
    # Se extraen los K vecinos mas cercanos (Menor distancia al valor
    medido)
    Kvecinos = distancias_ord[0:K]
```

Posteriormente se define un array donde cada posición corresponde a una clase, siendo clase 0 (decúbito supino), clase 1 (decúbito lateral izquierdo), clase 2 (decúbito

lateral derecho), clase 3 (sedestación) y clase 4 (semisedestación). A partir del array con la posición de los K vecinos mas cercanos (tres muestras de entrenamiento mas cercanas al dato a clasificar) se establecen condiciones para determinar la clase a la que pertenece. Mas adelante, se definen nombres para cada clase y se imprime el valor clasificado por consola, junto con la clase a la que pertenece.

```
# Identificación de la clase a la que pertenecen los K vecinos mas
cercanos
for i in range(K):
    vecino = Kvecinos[i]
    if (vecino >= 0 and vecino < 10):
        clases[0] += 1
    elif (vecino >= 10 and vecino < 20):
        clases[1] += 1
    elif (vecino >= 20 and vecino < 30):
        clases[2] += 1
    elif (vecino >= 30 and vecino < 40):
        clases[3] += 1
    else:
        clases[4] += 1

# Se obtiene el indice del array clase con mayor numero de vecinos,
dicho
# indice indica la clase a la que pertenece el valor medido,
clasificado
clasificado = str(np.argsort(clases)[4])
# Se establecen condiciones para asignar un nombre a cada clase
obtenida
if clasificado == '0':
    posicion = 'Decúbito supino'
elif clasificado == '1':
    posicion = 'Decúbito lateral izquierdo'
elif clasificado == '2':
    posicion = 'Decúbito lateral derecho'
elif clasificado == '3':
    posicion = 'Sedestación'
elif clasificado == '4':
    posicion = 'Semisedestación'
# Se imprime un mensaje indicando la clase a la que pertenece el
valor
print('El valor pertenece a la clase ' + clasificado)
```

C.3.7. Almacenamiento de datos clasificados en la base de datos

Finalmente, se abre nuevamente una conexión con la base de datos TESIS para insertar los datos clasificados en la tabla POSICION. Los valores insertados son un ID del dato, la fecha y hora al momento y el nombre de la postura a la que corresponde el dato clasificado. Al final del bucle existe una pausa en la ejecución del script que permite la clasificación de un nuevo dato aproximadamente cada segundo.

```
# Conexión a la base de datos TESIS
conexion2 = ConexionDB()
# Obtención de fecha y hora actual
timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
# Query SQL para insercion de datos clasificados en la tabla POSICION
query2 = """INSERT INTO `POSICIONES` (`ID`, `FECHA`, `POSICION`)
            VALUES (%s, %s, %s)"""
datos = (None, timestamp, posicion)
# Creación de cursor para ejecución de query
cursor2 = conexion2.cursor()
# Llamada al método execute para ejecutar la query diseñada
cursor2.execute(query2, datos)
# Llamada al método commit para afirmar cambios en la base de datos
conexion2.commit()
# Cierre de conexión con la base de datos
conexion2.close()

# Se ejecuta una pausa de 1 segundo al finalizar cada iteración
time.sleep(1)
```

C.4. Implementación de script PHP

El primer parámetro a configurar en el script PHP es el tiempo de ejecución, por defecto un bucle no puede ejecutarse indefinidamente, por lo que es necesario quitar el tiempo límite, a través de la siguiente instrucción:

```
//Desactiva el tiempo limite de ejecucion de script
set_time_limit(0);
```

Creamos la cadena de conexión hacia la base de datos TESIS, especificando el servidor, usuario y contraseña, tal y como se ve a continuación.

```
//Establecer cadena de conexión con la base de datos TESIS
$pdo=new PDO("mysql:dbname=TESIS;host=localhost","admin","1234");
```

Para determinar el tiempo que transcurre desde que el paciente está en una posición se configura la variable `start_time` con el tiempo del sistema a través de la función `microtime`.

```
//Inicializa variable para determinar tiempo transcurrido
$start_time = microtime(true);
```

Se extrae los registros más actuales ingresados en la base de datos, de las tablas DATOS y POSICIONES, y se separan las secciones a analizar como el nivel de batería y la posición del paciente. Posterior a esto se guarda el registro actual en una variable auxiliar para determinar si existe un cambio de posición en una siguiente iteración. De darse este cambio los contadores vuelven a inicializarse.

```
        //Extrae el ultimo registro de la tabla POSICIONES
        $statement=$pdo->prepare("SELECT FECHA, POSICION FROM
POSICIONES ORDER BY ID DESC LIMIT 0,1");

        //Extrae el ultimo registro de la tabla DATOS
        $statement2=$pdo->prepare("SELECT MEDICION FROM DATOS ORDER BY
ID DESC LIMIT 0,1");

        //Ejecutan las sentencias sql preparadas
        $statement->execute();
        $statement2->execute();

        //Asignar los resultados de las consultas a las variables de
resultados
        $results=$statement->fetchAll(PDO::FETCH_ASSOC);
        $results2=$statement2->fetchAll(PDO::FETCH_ASSOC);

        //Extrae unicamente el dato de POSICION del registro de la
tabla POSICIONES
        $pos = ($results[0]);
        $Paux =  $pos['POSICION'];

        //Extrae unicamente el dato de POSICION del registro de la
tabla POSICIONES
        $bat = ($results2[0]);
        $bat2 = $bat['MEDICION'];
        $bat2 = explode(",", $bat2);
        $bat3 = $bat2[2];

        //Si se da un cambio de posicion se inicializan los tiempos
        if ($pos2 != $Paux) {
                  $end_time = microtime(true);
                  $start_time = microtime(true);
        }
```

Luego de obtener las datos mencionados, estos son almacenados en el archivo `results.json`.

```
        //Abre archivo de resultados results.json para guardar datos
        $fp = fopen("/var/www/html/TESIS/results.json", "w");

        //Se ajusta el tiempo transcurrido al formato H:m:s
        $tiempo_tver = gmdate("H:i:s", $tiempo_t );

        //Se forma el arreglo de datos con la posicion, tiempo
transcurrido y nivel de batería.
        array_push($results,$tiempo_tver,$batfin);

        //Se ecriben los datos en el archivo results.json
        fwrite($fp, json_encode($results));

        //Se cierra el archivo
        fclose($fp);
```

Por último está la sección de código encargado del envío de alertas a través de telegram. La cadena de mensaje contiene el Bot, canal de difusión y texto a enviar.

```
        //Si el tiempo excede las dos horas se envia notificacion a
telegram
        if ($tiempo_t >= 7200 && $tiempo_t <= 7201 ){
                file_get_contents('https://api.telegram.org/
bot5261173171:AAElCfWZERpESxErciSkbPqSaZWNO-NAreI/sendMessage?chat_id=
@ALERTAUPP&text=ALERTA!! TIEMPO EXCEDIDO.. CAMBIE DE POSICION');
        }

        $batfin = 100*floatval($bat3)-300;

        if ($batfin <= 10 && $i == 0 ){
                file_get_contents('https://api.telegram.org/
bot5261173171:AAElCfWZERpESxErciSkbPqSaZWNO-NAreI/sendMessage?chat_id=
@ALERTAUPP&text=ALERTA!! NIVEL DE BATERIA BAJO');
                $i = 1;
        }
```

C.5. Implementación de código HTML

La interfaz del sistema está desarrollada en código html, tanto la pantalla de presentación, monitoreo y test de Braden.

El código de la interfaz de presentación es el siguiente:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Bootstrap Static Navbar</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/
    bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/
    bootstrap.bundle.min.js"></script>
</head>
<body>

        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-toggle="
    collapse" data-bs-target="#navbarText" aria-controls="navbarText" aria
    -expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarText">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                <li class="nav-item">
                    <a class="nav-link active" aria-current="page" href="
    #">HOME</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="Braden.html">BRADEN</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="monitoreo.html">RUTINA</a>
                </li>
                </ul>
            </div>
            </div>
        </nav>

<div style="background-image: url('img/fondo.jpg'); height: 88vh;">

        <div class="container pt-5">
        <p style="text-align:center; margin: top 5px;"><img src="img/
    logo_pequeño.png"  ></p>
        </div>
        <h5 class="display-6 text-center mt-5">
        SISTEMA DE MONITOREO PARA PREVENCIÓN DE ÚLCERAS POR PRESIÓN
        </h5>
        <div class="container mr-4">
```

```
            <h2 class="display-7 text-justify mt-5">
            Herramienta para monitorear la posición de pacientes
    encamados, usando sensores inerciales portátiles conformando una
    herramienta de
            control y vigilancia médica.
            </h2>
        </div>
</div>
<footer>
```

Por otro lado, para implementar el test de Braden usamos javascript de tal manera que la selección de cada respuesta no se repita, además asigna el nivel de riesgo según el valor total obtenido de cada ítem. El código usado está a continuación:

```
<!DOCTYPE html>
<html lang="en">
<head>

 <style>

   </style>

  <title> BRADEN</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="css/style.css" rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css
    /bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy
    +HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="
    anonymous">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.
    min.js"></script>
  <script type="text/javascript" src="https://www.gstatic.com/charts/
    loader.js"></script>


<script type="text/javascript">
  function seleccionar5(num){

  $("#ch1_"+num).on('change', function(ev){
      ev.preventDefault();

      $("#ch1_"+num).prop("checked", true);
      $("#ch2_"+num).prop("checked", false);
      $("#ch3_"+num).prop("checked", false);
      $("#ch4_"+num).prop("checked", false);
```

```
        $("#errores").html("Solo un check puede ser seleccionado");
        setTimeout(function(){
            $('#errores').html("");
        }, 2000);

    });
    $("#ch2_"+num).on('change', function(ev){
        ev.preventDefault();

        $("#ch2_"+num).prop("checked", true);
        $("#ch1_"+num).prop("checked", false);
        $("#ch3_"+num).prop("checked", false);
        $("#ch4_"+num).prop("checked", false);

        $("#errores").html("Solo un check puede ser seleccionado");
        setTimeout(function(){
            $('#errores').html("");
        }, 2000);

    });

    $("#ch3_"+num).on('change', function(ev){
        ev.preventDefault();

        $("#ch3_"+num).prop("checked", true);
        $("#ch1_"+num).prop("checked", false);
        $("#ch2_"+num).prop("checked", false);
        $("#ch4_"+num).prop("checked", false);

        $("#errores").html("Solo un check puede ser seleccionado");
        setTimeout(function(){
            $('#errores').html("");
        }, 2000);

    });
    $("#ch4_"+num).on('change', function(ev){
        ev.preventDefault();

        $("#ch4_"+num).prop("checked", true);
        $("#ch1_"+num).prop("checked", false);
        $("#ch2_"+num).prop("checked", false);
        $("#ch3_"+num).prop("checked", false);

        $("#errores").html("Solo un check puede ser seleccionado");
        setTimeout(function(){
            $('#errores').html("");
        }, 2000);
```

```
  });

  }
</script>
  <script type="text/javascript">
    function estados(){

      var resultado = document.getElementById('resu');
      //Pregunta 1
      var s_ch1_1 = document.getElementById('ch1_1').checked;
      var s_ch2_1 = document.getElementById('ch2_1').checked;
      var s_ch3_1 = document.getElementById('ch3_1').checked;
      var s_ch4_1 = document.getElementById('ch4_1').checked;

      let preg1 =[s_ch1_1,s_ch2_1,s_ch3_1,s_ch4_1];
      const indexp1 = preg1.findIndex(val => val === true)+1;
      // Pregunta 2
      var s_ch1_2 = document.getElementById('ch1_2').checked;
      var s_ch2_2 = document.getElementById('ch2_2').checked;
      var s_ch3_2 = document.getElementById('ch3_2').checked;
      var s_ch4_2 = document.getElementById('ch4_2').checked;

      let preg2 =[s_ch1_2,s_ch2_2,s_ch3_2,s_ch4_2];
      const indexp2 = preg2.findIndex(val => val === true)+1;

      // Pregunta 3
      var s_ch1_3 = document.getElementById('ch1_3').checked;
      var s_ch2_3 = document.getElementById('ch2_3').checked;
      var s_ch3_3 = document.getElementById('ch3_3').checked;
      var s_ch4_3 = document.getElementById('ch4_3').checked;

      let preg3 =[s_ch1_3,s_ch2_3,s_ch3_3,s_ch4_3];
      const indexp3 = preg3.findIndex(val => val === true)+1;



      // Pregunta 4
      var s_ch1_4 = document.getElementById('ch1_4').checked;
      var s_ch2_4 = document.getElementById('ch2_4').checked;
      var s_ch3_4 = document.getElementById('ch3_4').checked;
      var s_ch4_4 = document.getElementById('ch4_4').checked;

      let preg4 =[s_ch1_4,s_ch2_4,s_ch3_4,s_ch4_4];
      const indexp4 = preg4.findIndex(val => val === true)+1;

      // Pregunta 5
```

```
      var s_ch1_5 = document.getElementById('ch1_5').checked;
      var s_ch2_5 = document.getElementById('ch2_5').checked;
      var s_ch3_5 = document.getElementById('ch3_5').checked;
      var s_ch4_5 = document.getElementById('ch4_5').checked;

      let preg5 =[s_ch1_5,s_ch2_5,s_ch3_5,s_ch4_5];
      const indexp5 = preg5.findIndex(val => val === true)+1;

      // Pregunta 6
      var s_ch1_6 = document.getElementById('ch1_6').checked;
      var s_ch2_6 = document.getElementById('ch2_6').checked;
      var s_ch3_6 = document.getElementById('ch3_6').checked;
      var s_ch4_6 = document.getElementById('ch4_6').checked;

      let preg6 =[s_ch1_6,s_ch2_6,s_ch3_6,s_ch4_6];
      const indexp6 = preg6.findIndex(val => val === true)+1;
      var total = indexp1+indexp2+indexp3+indexp4+indexp5+indexp6;

      if (total<=12){
        resultado.value="Riesgo alto";
      } else if (total==13 || total==14){
        resultado.value="Riesgo moderado";
      } else if (total==15 || total==16){
        resultado.value="Riesgo bajo";
      }else if(total>=17){
        resultado.value="Sin riesgo";
      }

    }
  </script>


</head>
<body>

<div class="bg-image"  style="background-image: url('img/fondo.jpg'); ">

  <div style="margin-left:45px">



    <h1 class="display-5  text-center mt-1">
      TEST DE BRADEN
    </h1>

  <div class="m-3">
    <h4>P1. Percepción sensorial</h4>
```

```
</div>
<div class="col bg-white">

    <div class="col-md-3">
      <input type="checkbox" id="ch1_1" class="form-check-input"
      onclick="javascript:seleccionar5(1);">
      <label for="ch1_1" class="form-check-label "> Limitado
 completamente</label>
    </div>
    <div class="col-md-3">
      <input type="checkbox" id="ch2_1" class="form-check-input"
      onclick="javascript:seleccionar5(1);">
      <label for="ch2_1" class="form-check-label "> Muy limitado</label
>
    </div>
    <div class="col-md-3">
      <input type="checkbox" id="ch3_1" class="form-check-input"
      onclick="javascript:seleccionar5(1);">
      <label for="ch3_1" class="form-check-label "> Limitado levemente<
/label>
    </div>

    <div class="col-md-3">
      <input type="checkbox" id="ch4_1" class="form-check-input"
      onclick="javascript:seleccionar5(1);">
      <label for="ch4_1" class="form-check-label"> Sin impedimento</
label>
    </div>




</div>

<div class="m-3 ">
  <h4> P2. Exposición a la humedad</h4>
</div>

<div class="col bg-white">
  <div class="col-md-3">
    <input type="checkbox" id="ch1_2" class="form-check-input"
    onclick="javascript:seleccionar5(2);">
    <label for="ch1_2" class="form-check-label"> Constantemente húmeda<
/label>
  </div>
  <div class="col-md-3">
    <input type="checkbox" id="ch2_2" class="form-check-input"
```

```
      onclick="javascript:seleccionar5(2);">
      <label for="ch2_2" class="form-check-label"> Muy húmeda</label>
    </div>
    <div class="col-md-3">
      <input type="checkbox" id="ch3_2" class="form-check-input"
      onclick="javascript:seleccionar5(2);">
      <label for="ch3_2" class="form-check-label"> Ocasionalmente húmeda<
   /label>
    </div>

    <div class="col-md-3">
      <input type="checkbox" id="ch4_2" class="form-check-input"
      onclick="javascript:seleccionar5(2);">
      <label for="ch4_2" class="form-check-label"> Raramente húmeda</
   label>
    </div>
  </div>

  <div class="m-3 ">
    <h4>P3. Actividad física</h4>
  </div>


  <div class="col bg-white">
    <div class="col-md-3">
      <input type="checkbox" id="ch1_3" class="form-check-input"
      onclick="javascript:seleccionar5(3);">
      <label for="ch1_3" class="form-check-label "> Confinado en cama</
   label>
    </div>
    <div class="col-md-3">
      <input type="checkbox" id="ch2_3" class="form-check-input"
      onclick="javascript:seleccionar5(3);">
      <label for="ch2_3" class="form-check-label"> Confinado en silla</
   label>
    </div>
    <div class="col-md-3">
      <input type="checkbox" id="ch3_3" class="form-check-input"
      onclick="javascript:seleccionar5(3);">
      <label for="ch3_3" class="form-check-label"> Ocasionalmente camina<
   /label>
    </div>

    <div class="col-md-3">
      <input type="checkbox" id="ch4_3" class="form-check-input"
      onclick="javascript:seleccionar5(3);">
      <label for="ch4_3" class="form-check-label"> Camina frecuentemente<
```

```
 /label>
  </div>
</div>

<div class="m-3 ">
<h4>P4. Movilidad</h4>
</div>


<div class="col bg-white">
  <div class="col-md-3">
    <input type="checkbox" id="ch1_4" class="form-check-input"
    onclick="javascript:seleccionar5(4);">
    <label for="ch1_4" class="form-check-label"> Completamente inmóvil<
 /label>
  </div>
  <div class="col-md-3">
    <input type="checkbox" id="ch2_4" class="form-check-input"
    onclick="javascript:seleccionar5(4);">
    <label for="ch2_4" class="form-check-label"> Muy limitada</label>
  </div>
  <div class="col-md-3">
    <input type="checkbox" id="ch3_4" class="form-check-input"
    onclick="javascript:seleccionar5(4);">
    <label for="ch3_4" class="form-check-label"> Levemente limitada</
 label>
  </div>

  <div class="col-md-3">
    <input type="checkbox" id="ch4_4" class="form-check-input"
    onclick="javascript:seleccionar5(4);">
    <label for="ch4_4" class="form-check-label">Sin limitaciones</label
 >
  </div>
</div>

<div class="m-3">
  <h4>P5. Nutrición </h4>
</div>

<div class="col bg-white">

  <div class="col-md-3">
    <input type="checkbox" id="ch1_5" class="form-check-input"
    onclick="javascript:seleccionar5(5);">
    <label for="ch1_5" class="form-check-label"> Completamente
 inadecuada</label>
```

```
    </div>
    <div class="col-md-3">
      <input type="checkbox" id="ch2_5" class="form-check-input"
      onclick="javascript:seleccionar5(5);">
      <label for="ch2_5" class="form-check-label">Probablemente
   inadecuada</label>
    </div>
    <div class="col-md-3">
      <input type="checkbox" id="ch3_5" class="form-check-input"
      onclick="javascript:seleccionar5(5);">
      <label for="ch3_5" class="form-check-label">Adecuada</label>
    </div>

    <div class="col-md-3">
      <input type="checkbox" id="ch4_5" class="form-check-input"
      onclick="javascript:seleccionar5(5);">
      <label for="ch4_5" class="form-check-label"> Excelente</label>
    </div>
  </div>

  <div class="m-3 "><h4>P6. Fricción y roce</h4></div>

  <div class="col bg-white">
    <div class="col-md-3">
      <input type="checkbox" id="ch1_6" class="form-check-input"
      onclick="javascript:seleccionar5(6);">
      <label for="ch1_6" class="form-check-label">Presente</label>
    </div>
    <div class="col-md-3">
      <input type="checkbox" id="ch2_6" class="form-check-input"
```

Finalmente, la interfaz de monitoreo para asistencia médica está desplegada en HTML incluyendo de igual manera una sección de javascript, que ejecuta una función recursiva para presentar la información extraída de la base de datos. El código es el siguiente:

```
<!DOCTYPE HTML>
<html lang="en">

    <head>
        <meta http-equiv='cache-control' content='no-cache'>
        <meta http-equiv='expires' content='0'>
        <meta http-equiv='pragma' content='no-cache'>
        <title>MONITOREO</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale
   =1">
```

```
    <link href="css/style.css" rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/
dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1"
crossorigin="anonymous">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/
jquery.min.js"></script>

    <script type="text/javascript">

        function detener(){
            fetch('detener.php');

            document.getElementById("bot").disabled =false;
            document.getElementById("bot_D").disabled = true;
        }

        function comprobaciones(){
            document.getElementById("bot").disabled = true;
            document.getElementById("bot_D").disabled = false;
            fetch('crea.php');

            fetch('Contador.php');

            setInterval(function() {
            var JSON=$.ajax({
                url:"http://192.168.2.102/TESIS/results.json",
                dataType: 'json',
                async: false}).responseText;
            var Respuesta=jQuery.parseJSON(JSON);
            var x = document.getElementById('Pos');
            var tiem = document.getElementById('Tiempo');
                var bat = document.getElementById('Bateria');
            x.value = Respuesta[0].POSICION;
            tiem.value = Respuesta[1];
                bat.value = Respuesta[2];
            if ( x.value == "Decúbito supino") {
              document.getElementById('lugarfoto').src=("img/supino.
png");

            }  else if (x.value == "Semisedestación") {
              document.getElementById('lugarfoto').src=("img/
semisedestacion.png");

            } else if (x.value == "Sedestación") {
              document.getElementById('lugarfoto').src=("img/
sedestacion.png");
```

```
            }  else if (x.value == "Decúbito lateral derecho") {
              document.getElementById('lugarfoto').src=("img/
decubitoderecho.png");

            } else if (x.value == "Decúbito lateral izquierdo") {
              document.getElementById('lugarfoto').src=("img/
decubitoizquierdo.png");
            }

        }, 1000);
        }

    </script>
 </head>
 <body>
    <div class="bg-image"  style="background-image: url('img/fondo.
jpg');">
        <div class="container">
            <div class="container">
                <p style="text-align:center;" ><img src="img/
escudofinal.png" alt="Lo"  style="width:14vmax; height:6vmax; margin-
top: 3%;"></p>
            </div>
            <div class="container" style="margin-left: 2%;">
                    <div class="row">
                        <h3>POSICION</h3>
                    </div>
                    <div class="row justify-content-md-center">
                        <div class="col-sm">
                            <textarea name="Posi" id="Pos" cols="18"
rows="1" style="font-size: 12pt">...</textarea>
                        </div>

                        <div class="col-3" style="margin-left: 20%;
margin-right: 10%;">
                            <div class="banner-logo ">
                            <img id= "lugarfoto" src="img/supino.png"
 alt="Girl in a jacket" style="width:12vmax; height:10vmax;">
                            </div>
                        </div>

                    </div>
            </div>

            <div class="container" style="margin-left: 2%; margin-top
: 2%;">
```

```
                    <div class="row">
                        <h3>TIEMPO</h3>
                    </div>

                    <div class="row">
                        <div class="col-sm">
                            <textarea name="Tiempo" id="Tiempo" cols=
"10" rows="1" style="font-size: 12pt">...</textarea>
                        </div>
                        <div class="col-3" style="margin-left: 20%;
margin-right: 10%;">
                            <div class="banner-logo ">
                            <img id= "adv" src="img/suc.png" alt="
Girl in a jacket" style="width:10vmax;height:10vmax;">
                            </div>
                        </div>
                    </div>
              </div>

            <div class="container" style="margin-left: 2%; margin-top:
2%;">
                <div class="row">
                    <h3>BATERÍA</h3>
                </div>

                <div>
                    <div class="col-sm">
                        <textarea name="Bateria" id="Bateria" cols="
10" rows="1" style="font-size: 12pt">...</textarea>
                    </div>
                </div>
            </div>

                <div style="margin-left: 2%; margin-top:2%;">
                    <button type="button" class="btn btn-primary btn-
lg btn-block" id="bot" onclick="comprobaciones();">
                            INICIAR RUTINA <img src="img/play-circle
.svg" width="25" />
                     </button>

                     <button type="button" class="btn btn-danger btn-
lg btn-block" id="bot_D" onclick="detener();" disabled>
                            DETENER <img src="img/stop-circle.svg"
width="25" />
                    </button>
```

```
                        <button style="margin-left: 50%;" type="button"
    class="btn btn-default btn-lg" id="bot_R" onclick="location.href='
    index.html';" >
                            <img src="img/reg.svg" width="25" />
                            REGRESAR
                        </button>
                    </div>
            </div>
        </div>
    </body>
</html>
```

Bibliografía

[1] S. Coleman, J. Nixon, J. Keen, L. Wilson, E. McGinnis, C. Dealey, N. Stubbs, A. Farrin, D. Dowding, J. M. Schols *y otros*, "A new pressure ulcer conceptual framework," *Journal of advanced nursing*, vol. 70, num. 10, pp. 2222–2234, 2014.

[2] M. Benito, N. Moro, I. Prados *y otros*, "Prevención de las úlceras por presión en pacientes adultos," *Servicio Madrileño de Salud*, pp. 1–28, 2014.

[3] K. Spilsbury, A. Nelson, N. Cullum, C. Iglesias, J. Nixon, y S. Mason, "Pressure ulcers and their treatment and effects on quality of life: hospital inpatient perspectives," *Journal of advanced nursing*, vol. 57, num. 5, pp. 494–504, 2007.

[4] J. H. H. Valles, M. G. M. Monsiváis, M. Guzmán, G. Interial, y L. V. Arreola, "Cuidado de enfermería perdido en pacientes con riesgo o con úlceras por presión," *Revista Latino-Americana de Enfermagem*, vol. 24, 2016.

[5] E. O. Gushqui Carchi y J. A. Pazmiño López, "Actuación de enfermería en el manejo de pacientes con úlceras por presión en el área de especialidades clínicas de un hospital de la ciudad de guayaquil, desde mayo a septiembre de 2016." 2016.

[6] C. A. Russo, C. Steiner, y W. Spector, "Hospitalizations related to pressure ulcers among adults 18 years and older, 2006: Statistical brief# 64," 2011.

[7] I. Abubakar, T. Tillmann, y A. Banerjee, "Global, regional, and national age-sex specific all-cause and cause-specific mortality for 240 causes of death, 1990-2013: a systematic analysis for the global burden of disease study 2013," *Lancet*, vol. 385, num. 9963, pp. 117–171, 2015.

[8] X.-L. Zuo y F.-J. Meng, "A care bundle for pressure ulcer treatment in intensive care units," *International Journal of Nursing Sciences*, vol. 2, num. 4, pp. 340–347, 2015.

[9] S. Sprigle y S. Sonenblum, "Assessing evidence supporting redistribution of pressure for pressure ulcer prevention: a review," *J Rehabil Res Dev*, vol. 48, num. 3, pp. 203–13, 2011.

[10] F. C. Sánchez, "Cama para la prevención de úlceras por presión (upp)," *Enfermería Dermatológica*, vol. 2, num. 3, pp. 24–27, 2008.

[11] C. González, E. Cardiel, R. Muñoz, D. Villanueva, R. Urrutia, y P. Hernández, "Asiento hidráulico con movimiento para prevenir úlceras por presión," *Revista Mexicana de Ingeniería Biomédica*, vol. 27, num. 1, pp. 38–44, 2006.

[12] L. F. Cortés Torres *y otros*, "Validación de un dispositivo que establece y notifica la frecuencia de movimiento para prevenir úlceras por presión y caídas," Ph.D. dissertation, Universidad del Rosario, 2020.

[13] P. López-Casanova, J. Verdú-Soriano, M. Berenguer-Pérez, y J. Soldevilla-Agreda, "Prevención de las úlceras por presión y los cambios de postura. revisión integrativa de la literatura," *Gerokomos*, vol. 29, num. 2, pp. 92–99, 2018.

[14] E. B. Monroy, A. P. Rodríguez, M. E. Estevez, y J. M. Quero, "Fuzzy monitoring of in-bed postural changes for the prevention of pressure ulcers using inertial sensors attached to clothing," *Journal of Biomedical Informatics*, vol. 107, p. 103476, 2020.

[15] E. B. Monroy, D. Z. Romero, M. E. Estévez, F. Cruciani, I. Cleland, C. Nugent, y J. Medina-Quero, "Intelligent system for the prevention of pressure ulcers by monitoring postural changes with wearable inertial sensors," in *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 31, num. 1, 2019, p. 79.

[16] G. Cicceri, F. De Vita, D. Bruneo, G. Merlino, y A. Puliafito, "A deep learning approach for pressure ulcer prevention using wearable computing," *Human-centric Computing and Information Sciences*, vol. 10, num. 1, pp. 1–21, 2020.

[17] D. M. Minteer, P. Simon, D. P. Taylor, W. Jia, Y. Li, M. Sun, y J. P. Rubin, "Pressure ulcer monitoring platform—a prospective, human subject clinical study to validate patient repositioning monitoring device to prevent pressure ulcers," *Advances in wound care*, vol. 9, num. 1, pp. 28–33, 2020.

[18] A. P. Wai, S. F. Foo, W. Huang, J. Biswas, C.-C. Hsia, K. Liou, y P. Yap, "Lying posture classification for pressure ulcer prevention," *Journal of Healthcare Engineering*, vol. 1, num. 2, pp. 217–238, 2010.

[19] S. Caggiari, P. R. Worsley, S. L. Fryer, J. Mace, y D. L. Bader, "Detection of posture and mobility in individuals at risk of developing pressure ulcers," *Medical engineering & physics*, vol. 91, pp. 39–47, 2021.

[20] M. Heydarzadeh, M. Nourani, y S. Ostadabbas, "In-bed posture classification using deep autoencoders," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 3839–3842.

[21] U. Qidwai, S. Al-Sulaiti, G. Ahmed, A. Hegazy, y S. K. Ilyas, "Intelligent integrated instrumentation platform for monitoring long-term bedridden patients," in *2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)*. IEEE, 2016, pp. 561–564.

[22] D. Sen, J. McNeill, Y. Mendelson, R. Dunn, y K. Hickle, "A new vision for preventing pressure ulcers: wearable wireless devices could help solve a common-and serious-problem," *IEEE pulse*, vol. 9, num. 6, pp. 28–31, 2018.

[23] J. Gubbi, R. Buyya, S. Marusic, y M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, num. 7, pp. 1645–1660, 2013.

[24] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, y M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, num. 4, pp. 2347–2376, 2015.

[25] J. L. Mosquera Lorenzo, "Arduimu: Unidad de medición inercial basada en arduino," 2019.

[26] M. D. Pujos Yanzapanta, "Diseño e implementación de una unidad de medición inercial "imu" embebida en base a un sistema microcontrolado," B.S. thesis, Quito, 2016., 2016.

[27] L. M. d. P. Sánchez Galvis, "Prototipo basado en sensores inerciales para el seguimiento de la actividad física," 2019.

[28] H. F. Piñeros Lourenco *y otros*, "Evaluación del desarrollo y viabilidad de un sistema de detección de movimiento aplicable a pacientes de alto riesgo en caídas," Ph.D. dissertation, Universidad del Rosario, 2019.

[29] E. J. Carletti, "Comunicación-bus i2c," *Robots Argentina*, 2007.

[30] Z. T. Al-Dahan, N. K. Bachache, y L. N. Bachache, "Design and implementation of fall detection system using mpu6050 arduino," in *International Conference on Smart Homes and Health Telematics*. Springer, 2016, pp. 180–187.

[31] A. BANO, "Gesture control interface of a robot-car using raspberry pi," Ph.D. dissertation, University of Technology, 2019.

[32] G. Ferrer Mínguez, "Integración kalman de sensores inerciales ins con gps en un uav," 2009.

[33] D. Collins, "Motion basics: How to define roll, pitch, and yaw for linear systems," May 2020. [En línea]. Disponible: https://www.linearmotiontips.com/motion-basics-how-to-define-roll-pitch-and-yaw-for-linear-systems/

[34] J. D. Bernal Iñiguez, "Diseño, construcción e implementación de un sistema de captura de movimiento para análisis ergonómico de riesgo laboral de extremidades superiores," B.S. thesis, 2014.

[35] D. F. Pozo Espín, "Diseño y construcción de una plataforma didáctica para medir ángulos de inclinación usando sensores inerciales como acelerómetro y giroscopio," B.S. thesis, QUITO/EPN/2010, 2010.

[36] Í. S. Peña, "Sistema portable de sensores integrados para la adquisición de datos hacía el diseño de exoesqueletos," Ph.D. dissertation, Universidad de La Rioja, 2019.

[37] J. I. Pantoja Miño, "Interfaz gráfica para la adquisición de datos de la postura corporal de la muñeca por medio de sensores inerciales," B.S. thesis, 2021.

[38] J. M. Luna Jaramillo, "Desarrollo de un sistema de adquisición para la evaluación del balance corporal en base a la medida del movimiento del tronco."

[39] "Arduino core for ESP8266 WiFi chip," Jul. 2021, original-date: 2015-03-27T05:22:06Z. [En línea]. Disponible: https://github.com/esp8266/Arduino

[40] J. Ceja, R. Rentería, R. Ruelas, y G. Ochoa, "Módulo esp8266 y sus aplicaciones en el internet de las cosas," *Revista de Ingeniería eléctrica*, vol. 1, num. 2, pp. 24–36, 2017.

[41] R. K. Kodali y K. S. Mahesh, "A low cost implementation of mqtt using esp8266," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2016, pp. 404–408.

[42] T. Thaker, "Esp8266 based implementation of wireless sensor network with linux based web-server," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE, 2016, pp. 1–5.

[43] R. K. Kodali y K. S. Mahesh, "Low cost ambient monitoring using esp8266," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 2016, pp. 779–782.

[44] "WeMos D1 mini ESP8266 WiFi." [En línea]. Disponible: https://naylampmechatronics.com/espressif-esp/291-wemos-d1-mini-esp8266-wifi.html

[45] Á. Espejo Muñoz, "Diseño de servidor para sistemas distribuidos sobre dispositivos raspberry pi," 2019.

[46] S. E. Princy y K. G. J. Nigel, "Implementation of cloud server for real time data storage using raspberry pi," in *2015 Online International Conference on Green Engineering and Technologies (IC-GET)*. IEEE, 2015, pp. 1–4.

[47] A. D. Espinoza Altamirano, "Desarrollo de un prototipo de iot basado en el módulo wifi y servidor web local en la plataforma raspberry-pi enfocado para personas con capacidades especiales de movilidad." B.S. thesis, Quito, 2021, 2021.

[48] D. D. Agurto Valverde, "Integración de un sistema de adquisición de datos mediante el uso de un arduino mega y raspberry pi 3 como servidor web y base de datos," 2020.

[49] R. A. Rodríguez, P. M. Vera, D. A. Giulianelli, y P. Cammarano, "Implementación con raspberry pi de un servidor portátil de contenidos," in *XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017).*, 2017.

[50] S. S. Chamorro Pinto, "Puesta en marcha de una estación meteorológica integrada a un servidor web en un raspberry pi de características óptimas," B.S. thesis, 2015.

[51] C. W. Zhao, J. Jegatheesan, y S. C. Loon, "Exploring iot application using raspberry pi," *International Journal of Computer Networks and Applications*, vol. 2, num. 1, pp. 27–34, 2015.

[52] A. Thapliyal y C. Kumar, "Development of data acquisition console and web server using raspberry pi for marine platforms," *International Journal of Information Technology and Computer Science*, vol. 8, pp. 46–53, 2016.

[53] B. Vaidya, A. Patel, A. Panchal, R. Mehta, K. Mehta, y P. Vaghasiya, "Smart home automation with a unique door monitoring system for old age people using python, opencv, android and raspberry pi," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2017, pp. 82–86.

[54] H. Guerra, A. Cardoso, V. Sousa, J. Leitão, V. Graveto, y L. M. Gomes, "Demonstration of programming in python using a remote lab with raspberry pi," in *2015 3rd Experiment International Conference (exp. at'15)*. IEEE, 2015, pp. 101–102.

[55] J. Potter, "What is a LAMP stack?" May 2021. [En línea]. Disponible: https://www.liquidweb.com/kb/what-is-a-lamp-stack/

[56] B. Chavarría Neira y E. Gudiño De La, "Implementación de un servidor web y un diseño de una página utilizando herramientas de software libre para el dispensario sagrada familia de la ciudad de guayaquil." B.S. thesis, 2017.

[57] A. Banks y R. Gupta, "Mqtt version 3.1. 1, vol. 29," *OASIS Standard, Burlington, MA, USA*, 2014.

[58] B. Ullas, S. Anush, J. Roopa, y M. G. Raju, "Machine to machine communication for smart systems using mqtt," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, num. 3, pp. 8242–8248, 2014.

[59] R. K. Kodali y S. Soratkal, "Mqtt based home automation system using esp8266," in *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. IEEE, 2016, pp. 1–5.

[60] R. K. Kodali y A. Valdas, "Mqtt based monitoring system for urban farmers using esp32 and raspberry pi," in *2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)*. IEEE, 2018, pp. 395–398.

[61] D. Eridani y E. D. Widianto, "Performance of sensors monitoring system using raspberry pi through mqtt protocol," in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 2018, pp. 587–590.

[62] K. Chooruang y P. Mangkalakeeree, "Wireless heart rate monitoring system using mqtt," *Procedia Computer Science*, vol. 86, pp. 160–163, 2016.

[63] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, vol. 1, num. 2011, pp. 1–11, 2011.

[64] K. Grgić, I. Špeh, y I. Heđi, "A web-based iot solution for monitoring data using mqtt protocol," in *2016 international conference on smart systems and technologies (SST)*. IEEE, 2016, pp. 249–253.

[65] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, y J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," *Transaction on IoT and Cloud computing*, vol. 3, num. 1, pp. 11–17, 2015.

[66] G. C. Hillar, *MQTT Essentials-A lightweight IoT protocol*. Packt Publishing Ltd, 2017.

[67] "Eclipse Mosquitto," Ene. 2018. [En línea]. Disponible: https://mosquitto.org/

[68] R. Fonnegra, G. Goez, y A. Tobón, "Estimación de orientación de un vehículo aéreo no modelado usando fusión de sensores inerciales y aprendizaje de máquina," *Revista Iberoamericana de Automática e Informática.*, vol. 16, num. 4, pp. 415–422, 2019.

[69] A. Martínez Parra *y otros*, "Estudio de sistemas inerciales en el seguimiento de terapias rehabilitadoras basadas en machine learning," 2021.

[70] J. Sora Cárdenas, "Evaluación del riesgo de caídas en adultos mayores con neuropatía diabética periférica," 2018.

[71] F. B. González, "Cuidados familiares," *¿ Qué es la Esclerosis Múltiple? Cuidados familiares para personas gravemente afectadas de Esclerosis Múltiple*, p. 9.

[72] O. J. Patiño, H. A. Aguilar, y A. L. Belatti, "Úlceras por presión: cómo prevenirlas," *Rev. Hosp. Ital. B. Aires*, vol. 38, num. 1, pp. 40–46, 2018.

[73] M. Lima-Serrano, M. González-Méndez, C. Martín-Castaño, I. Alonso-Araujo, y J. S. Lima-Rodríguez, "Validez predictiva y fiabilidad de la escala de braden para valoración del riesgo de úlceras por presión en una unidad de cuidados intensivos," *Medicina Intensiva*, vol. 42, num. 2, pp. 82–91, 2018.

[74] R. Serrano, "ESTADO DE CARGA DE UNA BATERÍA," Mar. 2018. [En línea]. Disponible: https://tritec-intervento.cl/estado-de-carga-de-una-bateria/

[75] R. P. Ltd, "Buy a Raspberry Pi 3 Model B+." [En línea]. Disponible: https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/