



**UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

CARRERA DE INGENIERÍA MECATRÓNICA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN MECATRÓNICA**

TEMA:

**“ALGORITMO DE DETECCIÓN DE CAÍDAS PARA PERSONAS DE LA
TERCERA EDAD”**

AUTOR: BRYAN ALEXANDER RIVADENEIRA POZO

DIRECTOR: Ph.D. DAVID ALBERTO OJEDA PEÑA

Ibarra-Ecuador

2023



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004386197		
APELLIDOS Y NOMBRES:	Rivadeneira Pozo Bryan Alexander		
DIRECCIÓN:	Juana Atabalipa y Hernan Gonzales de Saa		
EMAIL:	barivadeneirap@utn.edu.ec		
TELÉFONO FIJO:	618-217	TELÉFONO MÓVIL:	0988179985

DATOS DE LA OBRA	
TÍTULO:	Algoritmo de detección de caídas para personas de la tercera edad.
AUTOR (ES):	Rivadeneira Pozo Bryan Alexander
FECHA:	28/04/2023
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN MECATRÓNICA
ASESOR /DIRECTOR:	PhD. David Alberto Ojeda Peña

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 28 días del mes de abril de 2023

EL AUTOR:

A handwritten signature in blue ink is written over a horizontal dotted line. The signature is stylized and appears to be 'Rivadeneira J.'. The name 'RIVADENEIRA J.' is printed in blue capital letters directly beneath the signature.

Rivadeneira Pozo Bryan Alexander

3. CERTIFICACIÓN.

En calidad de tutor del presente Trabajo de Grado titulado: “Algoritmo de detección de caídas para personas de la tercera edad”, certifico que fue desarrollado por el señor Rivadeneira Pozo Bryan Alexander, bajo mi supervisión.

A handwritten signature in blue ink, consisting of several loops and a long horizontal stroke extending to the right, positioned above a dotted line.

PhD. David Ojeda

DIRECTOR DEL PROYECTO

DEDICATORIA

Este proyecto va dedicado a toda mi familia en especial a mis padres, hermanos y abuelita que siempre estuvieron apoyando con sus gestos de cariño durante toda mi vida, son y siempre serán mi fuente de inspiración y motivación para continuar superándome cada día.

También agradezco a todos mis amigos que formaron parte de mi etapa como estudiante y compartieron su alegrías y tristezas durante todos estos años.

Gracias también a mis tutores por su tiempo, apoyo y palabras de motivación durante todo el desarrollo de este trabajo.

AGRADECIMIENTOS

Gracias a mis familiares, tutores y amigos que apoyaron con su tiempo para el desarrollo de este proyecto, gracias a la Universidad Técnica del Norte por su gestión brindo las herramientas necesarias para el desarrollo de este trabajo.

RESUMEN

En el Ecuador la incidencia anual de caídas se encuentra entre el 25% y 50% en personas de la tercera edad lo que posiciona este accidente como la quinta causa de muerte y a nivel mundial según la OMS se posiciona como la segunda. Por lo visto anteriormente este accidente ha sido motivo del desarrollo de diferentes sistemas que detectan una caída, por lo tanto se realizó un estudio de los mismos, donde se obtuvieron los parámetros y requerimientos para el diseño del algoritmo de detección de caídas, este sistema usa la cinemática de una caída y umbrales, tiene como propósito informar a la persona pertinente de que ha ocurrido este accidente mediante un correo electrónico, para su validación se diseñó y construyó un sistema embebido portátil compuesto principalmente por dos microcontroladores y dos IMUs (unidad de medición inercial) cada uno de estos dispositivos son ubicados cerca del centro de gravedad y la rodilla, para evaluar el desempeño del algoritmo se usó una matriz de confusión donde se seleccionó actividades comunes, caídas flexionando y sin flexionar las rodillas. Finalmente, los resultados obtenidos a partir de las pruebas experimentales dieron como resultado un tiempo entre la activación de alarma y el impacto en promedio de 99,8ms para las caídas flexionando las rodillas y de 116,4 ms para caídas con las piernas extendidas, sensibilidad de 96,43%, especificidad de 97,86% y precisión del 97,14%.

Palabras clave: Algoritmo, Detección de caídas, Umbrales, Portable.

ABSTRACT

In Ecuador, the annual incidence of falls is between 25% and 50% in elderly people, which positions this accident as the fifth cause of death and worldwide, according to the OMS, it is positioned as the second. As previously seen, this accident has been the reason for the development of different systems that detect a fall, therefore a study of them was carried out, where the parameters and requirements for the fall detection algorithm were acquired, this system uses the kinematics of a fall and thresholds, The purpose of the algorithm is to send an alert by email to the person in charge of user care, for its validation a portable embedded system was designed and built, mainly consisting of two microcontrollers and two IMUs (Unit of inertial measurement) these devices are located near the center of gravity and the knee, to evaluate the performance of the algorithm a confusion matrix was used where common activities, falls flexing and without flexing the knees were selected, Finally, the results obtained of experimental tests resulted in an average time between alarm activation and impact of 99.8ms for falls with knee flexion and 116.4ms for falls with extended legs, sensitivity of 96.43%, specificity of 97.86% and precision of 97.14%.

Keywords: Algorithm, Falls Detection, Thresholds, Portable.

ÍNDICE GENERAL

CAPITULO I	14
1.1 Problema.....	14
1.2 Objetivos.....	14
1.2.1 Objetivo General:.....	14
1.2.2 Objetivos específicos:.....	14
1.3 Justificación.....	15
1.4 Alcance	15
CAPITULO II.....	16
2 Marco Referencial.....	16
2.1 Antecedentes.....	16
2.2 Marco Teórico	18
2.2.1 Biomecánica.....	18
2.2.1.1 Cinemática.....	18
2.2.1.2 Centro de masa (CM) y Centro de gravedad (CG).....	18
2.2.1.3 Base de sustentación (BDS)	18
2.2.1.4 Equilibrio y Estabilidad.....	18
2.2.1.5 Línea de gravedad	18
2.2.1.6 Ángulo de caída.....	19
2.2.2 Tipos de sistemas de detección de caídas	19
2.2.2.1 No portátiles.	20
2.2.2.2 Portátiles.....	23
2.2.2.3 Predictivo	33
2.2.3 Sistema de adquisición de datos: Acondicionamiento de señal.....	34
2.2.3.1 Tipos de filtros	34
2.2.4 Protocolo simple de transferencia de correo.....	35
CAPÍTULO III.....	36
3 Marco metodológico	36
3.1 Modelo de la Investigación.....	36
3.2 Diseño de la Investigación.....	36
3.2.1 Investigación acerca de los algoritmos de detección de caídas.	36
3.2.2 Diseño de un algoritmo de detección de caídas.....	36

3.2.3	Evaluación del funcionamiento del algoritmo.	37
CAPÍTULO IV.....		39
4	Resultados y análisis	39
4.1	Especificaciones del algoritmo a diseñar.....	39
4.2	Selección del método de detección de caída.	40
4.3	Selección de tipo de algoritmo	42
4.4	Metodología para el desarrollo del algoritmo.....	44
4.4.1	Selección de componente para el sistema embebido para la validación del algoritmo.	45
4.4.1.1	Selección de sensor:	45
4.4.1.2	Selección de filtro:	45
4.4.1.3	Selección de microcontrolador:.....	46
4.4.1.4	Selección de baterías	47
4.4.2	Diseño del sistema embebido para la adquisición de datos y validación del algoritmo.	47
4.4.3	Diseño del algoritmo.....	48
4.4.4	Diseño del sistema de sujeción para el sistema embebido.....	52
4.4.5	Resultados del algoritmo.....	52
4.4.6	Tiempo de espera.	54
4.5	Diseño del algoritmo para detectar caídas flexionando las rodillas.	55
4.5.1	Diseño del sistema embebido para la rodilla.	55
4.5.2	Selección de umbrales de desequilibrio y caída libre para la rodilla.	56
4.5.3	Algoritmo para las caídas flexionando la rodilla.	58
4.5.4	Diseño de la red inalámbrica usando ESP-Now	61
4.5.5	Diseño del sistema de sujeción de los sensores.	61
4.5.6	Adquisición de datos para obtener el tiempo de espera.	62
4.5.7	Matriz de confusión	64
4.5.8	Ajustes y correcciones.	66
4.5.9	Implementación del protocolo simple de transferencia de correo.	69
CONCLUSIONES		71
RECOMENDACIONES.....		72
REFERENCIAS.....		73
ANEXOS		76

ÍNDICE DE FIGURAS

Figura 2.1 Sistema de protección usable y controlable (a) Bolsa de aire, (b) Deposito y accionamiento del aire comprimido y (c) Diagrama de bloque del sistema de control que acciona el mecanismo [7].....	16
Figura 2.2 Actuador para inflar la bolsa de aire.[8].....	17
Figura 2.3 Centro de gravedad y su proyección en la base de sustentación en marcha y estático la línea segmentada indica la línea de gravedad [13].	18
Figura 2.4 Centro de gravedad, línea de gravedad, ángulo de caída y proyección del centro de gravedad sobre la base de sustentación en una actividad deportiva [13].....	19
Figura 2.5 Tipos de sistemas detectores de caídas [14], [15].	20
Figura 2.6 Resultado de la extracción humana [16].	21
Figura 2.7 Técnica de histograma de proyección [16].....	21
Figura 2.8 Algoritmo de detección de caídas basado en cámaras [16].....	21
Figura 2.9 Sistema de detección de caídas usando sensor infrarrojo [17].....	22
Figura 2.10 Diagrama de flujo que muestra los pasos que sigue este sistema de detección de caídas y el funcionamiento del algoritmo basado en umbrales [18].....	23
Figura 2.11 Diagrama de bloque de las partes del sistema embebido [19].....	24
Figura 2.12 Diagrama de flujo del sistema de detección de caídas [19].....	25
Figura 2.13 (a) Diagrama de cajas de la velocidad máxima y mínima y (b) máximo y mínimo de desplazamiento para cada actividad. Caída parada a sentado (PAS caída) las demás caídas son agrupadas juntas y asignadas a caída normal (Normal caída) [19].....	25
Figura 2.14 Diagrama de bloque del sistema [22].	27
Figura 2.15 Arquitectura propuesta [23].....	28
Figura 2.16 Diagrama de flujo de las fases y características cinemáticas de una caída [15].....	32
Figura 2.17. Simulación de una caída hacia adelante mostrando cada uno de los parámetros previamente descritos en las ecuaciones [15].	32
Figura 2.18 Circuito filtro pasa bajo RC serie [25].	34
Figura 2.19 Circuito filtro pasa alto RC serie [25].	35
Figura 4.1 Fotogramas de una cámara de video de accidentes de caída en personas de la tercera edad [31].	44
Figura 4.2 Sensor MPU 6050 [32].	45

Figura 4.3 a) Microcontrolador ESP-32 WROOM [33]. b) Microcontrolador ESP el 8266 mini D1 [34].	46
Figura 4.4 Batería de li-on.	47
Figura 4.5 Esquema de las conexiones del dispositivo uno.	48
Figura 4.6 Soporte para el sensor MPU – 6050.	49
Figura 4.7 Señal filtrada del sensor MPU-6050 usando el filtro complementario de media móvil.	49
Figura 4.8 Diagrama de flujo de detección de una caída.	51
Figura 4.9 Sistema de sujeción para el sistema embebido.	52
Figura 4.10 Muestra de una caída sin flexionar las rodillas.	54
Figura 4.12 Muestra de una actividad de caída flexionando las rodillas.	56
Figura 4.13 Muestra de aceleración de la actividad de agacharse apoyando las rodillas.	57
Figura 4.14 Muestra de una actividad de caída flexionando las rodillas	58
Figura 4.15 Diagrama de una caída.	59
Figura 4.16 Diagrama de flujo para el dispositivo uno y dos.	60
Figura 4.17 Configuración de red.	61
Figura 4.18 Estructura de sujeción del sistema embebido y el cuerpo.	61
Figura 4.19 Ubicación y envío de datos de los dispositivos para la adquisición de datos.	62
Figura 4.20 Muestra de caída sin flexionar la rodilla.	63
Figura 4.21 Muestra de caída flexionando las rodillas.	64
Figura 4.22 Actividades con un mayor índice de falsos positivos a) Agacharse con apoyo de rodilla, b) Subir/bajar escaleras, c) Acostarse y levantarse.	65
Figura 4.23 Actividad de agacharse.	66
Figura 4.24 Cinemática de la actividad de acostarse.	67
Figura 4.25 Algoritmo de detección de caídas.	68
Figura 4.27 Programación en el IDE de Arduino de la configuración del protocolo SMTP.	70
Figura 4.28 Programación en el IDE de Arduino de la función que se encarga de enviar el correo.	70
Figura 4.29 Visualización del correo recibido.	70

ÍNDICE DE TABLAS

Tabla 2.1. Descripciones de las partes que conforman el sistema embebido.	26
Tabla 2.2 Resultados del algoritmo de detección de caídas [21].	27
Tabla 2.3. Valores mínimos y máximos registrados en actividades de la vida diaria y caídas [15].	30
Tabla 2.4 Descripción de la configuración y partes que conforman el sistema embebido del sistema.....	33
Tabla 4.1 Ventajas y desventajas de los sistemas de caída portátiles, no portátiles y predictivos [18], [19], [21], [22], [23], [15].	41
Tabla 4.2 Identificación de las características de cada tipo de algoritmo [18], [19], [21], [22], [23], [15]	43
Tabla 4.3 Especificaciones del sensor MPU – 6050.....	45
Tabla 4.4 Especificaciones del microcontrolador ESP – 32 y ESP 8266	47
Tabla 4.5 Matriz de confusión del algoritmo de detección de caídas sin flexionar las rodillas. ..	53
Tabla 4.6 Resultados del algoritmo de detección de caídas sin flexionar las rodillas.	54
Tabla 4.7 Muestras registradas.	55
Tabla 4.8 Aceleración mínima y máxima de cinco muestras de la actividad de agacharse.	57
Tabla 4.9 Aceleración mínima y máxima de cinco muestras de la actividad de caída flexionando las rodillas.	58
Tabla 4.10 Tabla de resultados de las muestras de caídas sin flexionar las rodillas.....	63
Tabla 4.11 Tabla de resultados de las muestras de caídas flexionando las rodillas.....	64
Tabla 4.12 Matriz de confusión	65
Tabla 4.13 Matriz de confusión.	69

CAPITULO I

1.1 Problema

Ecuador es un país en el que la esperanza de vida de la población está aumentando. Según el informe mensual de gestión del servicio de población adulta mayor realizado por el ministerio de inclusión económica y social (MIES), afirma que, entre los 7 países de la región, Ecuador ocupa el cuarto puesto con 28 adultos mayores por cada 100 menores de los cuales 132.365 viven solos [1].

Adicionalmente, la guía ecuatoriana de atención para el adulto mayor, la incidencia anual de caídas en ancianos entre 65 a 70 años es de 25% y se estima que pueda llegar entre el 35% a 45% al tener edades más avanzadas (80 a 85 años). Así mismo afirman que hasta el 30% de los ancianos se caen una vez al año, sobre todo las mujeres mientras que en los adultos mayores institucionalizados se eleva hasta el 50%; donde el 17% presenta consecuencias graves. Finalmente reportan que las caídas son la 5ª causa de muerte en las personas adultas mayores[2].

Por otro lado, el síndrome poscaída afecta a más del 80% de las personas que han sufrido una caída, además de lesiones físicas (heridas, contusiones y fracturas de cadera principalmente) lo que conlleva un inmenso gasto de recursos económicos, humanos y sociosanitarios a los que la sociedad debe hacer frente, iniciándose el proceso de discapacidad en el anciano [3]. Además, las dos terceras partes de los ancianos que se caen sufrirán una nueva caída en los siguientes seis meses[4].

Por lo tanto, el uso de un algoritmo que detecte que una persona se está cayendo permitirá servir como base para el desarrollo de soluciones que protejan al adulto mayor de sufrir consecuencias irreversibles. Además, podrá notificar del accidente a sus cuidadores o servicios de emergencia cercanos para reducir las complicaciones que conlleva no ser atendidos de forma inmediata.

1.2 Objetivos

1.2.1 Objetivo General:

- Desarrollar un algoritmo de detección de caídas acorde a las condiciones de vida comunes del adulto mayor.

1.2.2 Objetivos específicos:

- Investigar acerca de los algoritmos existentes de detección de caídas.
- Diseñar un algoritmo de detección de caídas.
- Validar el funcionamiento del algoritmo.

1.3 Justificación

Los adultos mayores son uno de los sectores más vulnerables de la sociedad pues al verse disminuida parte de sus funciones motrices, psicológicas y presentar algunas patologías propias de la edad que no son posibles corregir por un simple tratamiento, necesitan de apoyos y cuidados que anteriormente eran solventadas por la persona que se quedaba en el hogar [5].

El uso del algoritmo capaz de detectar caídas ayudará a reducir los índices de muerte por esta problemática y evitará las complicaciones que conlleva que el adulto mayor de no ser atendido de forma inmediata después del accidente.

Adicionalmente, servirá como referencia a futuras investigaciones para desarrollar un sistema de amortiguación de caídas para proteger de lesiones graves en las zonas más importantes del cuerpo como la cabeza, tronco y cadera evitando las consecuencias económicas, físicas y psicológicas que conlleva la caída en una persona adulta mayor.

1.4 Alcance

En el presente trabajo se estudiarán los distintos métodos que existen actualmente para la detección de caídas, identificando las características de cada uno de ellos, para luego desarrollar uno tomando en cuenta las características y criterios obtenidos.

El sistema para desarrollar constará de un algoritmo capaz de monitorear constantemente los movimientos del usuario mediante sensores y deberá diferenciar entre una caída de un movimiento común (sentarse, acostarse, caminar).

Finalmente, se simulará el algoritmo con pruebas experimentales para determinar su precisión y tiempo de respuesta.

CAPITULO II.

2 Marco Referencial

2.1 Antecedentes

La caída es un desafío para las personas de la tercera edad que reduce la esperanza de vida de esta parte de la población [6]. Constantemente se desarrolla tecnología para dar solución a las necesidades. En detección de caídas se han estudiado e implementado varios sistemas estos sistemas en su mayoría usan cámaras, sensor Kinect y sensores ubicados en el cuerpo.

En China se desarrolló un sistema de protección usable como alternativa de solución. Propusieron un diseño basado en un sistema de control como se muestra en la figura 2.1 (b) indica un diagrama de bloques de cómo está constituido, monitorea el estado del usuario mediante un sensor y en caso de detectar una caída a través de un algoritmo que detecta la caída, activa el mecanismo mostrado en la figura 2.1 (a), se encarga de abrir el depósito de aire comprimido que está conectado a la bolsa de aire figura 2.1 (c), con el propósito de amortiguar el impacto de la caída evitando lesiones en la zona de la cadera (ver figura 2.1) [7].

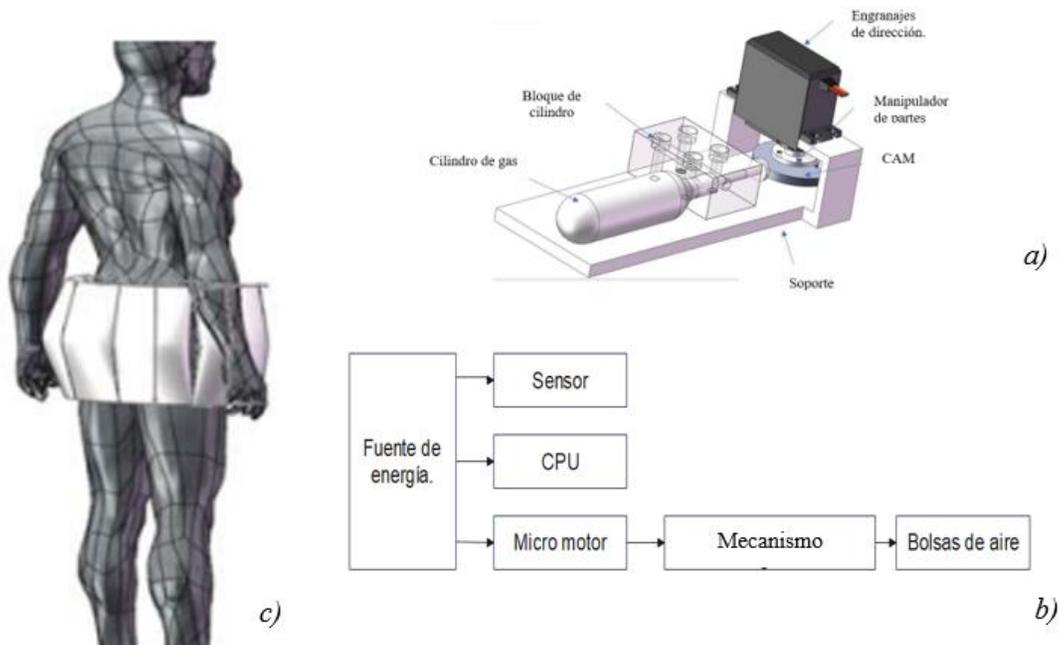


Figura 2.1 Sistema de protección usable y controlable (a) Bolsa de aire, (b) Depósito y accionamiento del aire comprimido y (c) Diagrama de bloques del sistema de control que acciona el mecanismo [7].

Se han desarrollado sistemas más sofisticados, monitorea el estado de la persona y al detectar la caída, envía un mensaje de auxilio, ubicación y amortigua el impacto. Incorpora un mecanismo con dos cilindros de aire comprimido que ayuda a reducir el tiempo de inflado de la bolsa de aire (ver figura 2.2) [8].

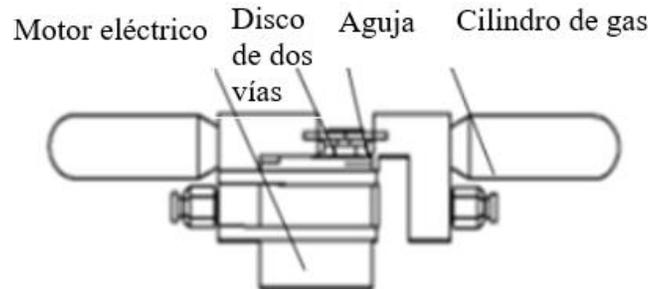


Figura 2.2 Actuador para inflar la bolsa de aire.[8].

El uso de cámaras para detectar caídas tiene una tasa de reconocimiento mayor del 90%, es logrado gracias a la visión artificial y aprendizaje de máquina, pero el uso de este tipo se ve limitado en el área de reconocimiento y muchas veces es rechazado por el usuario debido a que compromete su privacidad [9].

Para mejorar el sistema basado en cámaras se diseñaron otros métodos que incorporan el sensor Kinect, un sensor capaz de mapear en 3D el entorno mediante un emisor de infrarrojos y una cámara de profundidad, esta adición al sistema permite reconocer mejor la ubicación del usuario en el entorno, aumentado la tasa de reconocimiento y mantener la privacidad de la persona de la tercera edad [10].

A pesar de que las caídas son eventos que ocurren con poca frecuencia y aumenta con respecto a la edad de la persona, sus consecuencias son graves en personas de la tercera edad, logrando así convertirse en un problema de la salud pública muy importante lo que dio paso a muchas investigaciones para hacer frente a la misma y a su vez artículos de revisión que tienen como propósito proporcionar una visión general de los estudios actuales relacionados con la detección de caídas e inspirar a los investigadores de este campo [11].

2.2 Marco Teórico

2.2.1 Biomecánica

2.2.1.1 Cinemática

La ciencia que estudia el movimiento de un cuerpo sin tener en cuenta qué lo causa. La cinemática describe la trayectoria, la velocidad y la aceleración del movimiento de un cuerpo con referencia al centro de gravedad, estos movimientos pueden ser lineales, angulares o combinados [12].

2.2.1.2 Centro de masa (CM) y Centro de gravedad (CG)

El centro de masa se define como la posición promedio de todas las partículas de masa que forman el cuerpo [12] y se define al centro de gravedad como la ubicación de la resultante de todas las fuerzas de gravedad que actúan sobre un cuerpo, estas dos posiciones se localizan en el mismo punto en el cuerpo humano [13].

2.2.1.3 Base de sustentación (BDS)

Es el área que se forma al unir los puntos de apoyo más distantes, en las personas se conforman por los bordes externos de los dos pies y todo lo que hay entre ellos como se indica en la figura 2.3 [13].

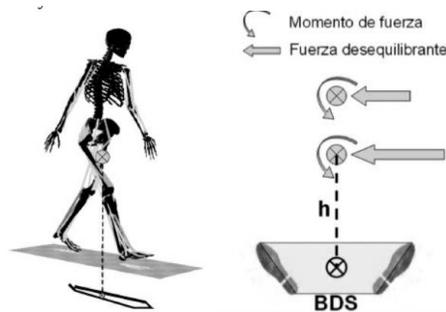


Figura 2.3 Centro de gravedad y su proyección en la base de sustentación en marcha y estático la línea segmentada indica la línea de gravedad [13].

2.2.1.4 Equilibrio y Estabilidad

En el ámbito de la biomecánica se denomina equilibrio a la capacidad de la persona de mantener su propio u otro cuerpo en una posición controlada, la estabilidad es la capacidad de la persona para mantener el equilibrio es decir que puede mantener su posición a pesar de perturbaciones externas [12].

2.2.1.5 Línea de gravedad

Es una línea normal al piso que pasa por el centro de gravedad y proyecta el mismo sobre el suelo esta proyección nos ayuda a determinar si la persona está o no en equilibrio si la proyección está

dentro del BDS se puede decir que está en equilibrio estable, mientras este cerca de un borde de la BDS se dice que está en equilibrio inestable y si queda fuera está en desequilibrio [13].

2.2.1.6 Ángulo de caída

Este ángulo se forma entre un plano vertical que pasa por el borde de la base de sustentación y otro plano que pasa por el mismo borde y el centro de gravedad como se muestra en la figura 2.4 si el ángulo formado es negativo significa que la proyección del CG se ha salido de la BDS [13].

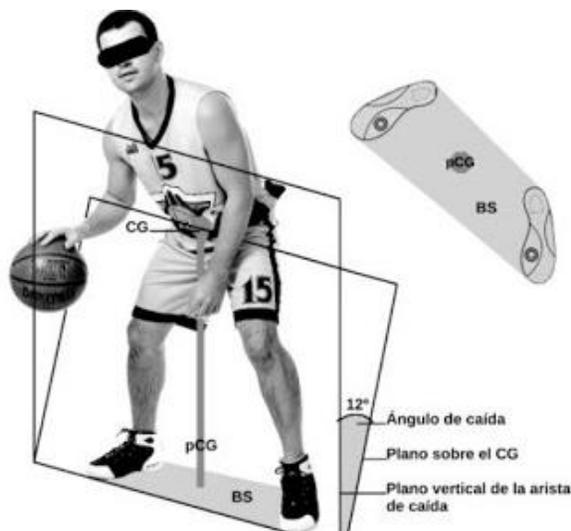


Figura 2.4 Centro de gravedad, línea de gravedad, ángulo de caída y proyección del centro de gravedad sobre la base de sustentación en una actividad deportiva [13].

2.2.2 Tipos de sistemas de detección de caídas

Las tecnologías utilizadas para detectar las caídas se los puede agrupar en tres que se diferencia según si el sistema es portátil o no y otro método usado en la medicina que para este caso se lo agrupa como predictivo [14].

Además, cada tipo de sistema también se subclasifican según muestra la figura 2.5 para la categoría de no portátiles, según el tipo de sensores que usa [14], para los dispositivos portátiles según el algoritmo que usan [15] y para los métodos predictivos esta la escala de Tinetti.

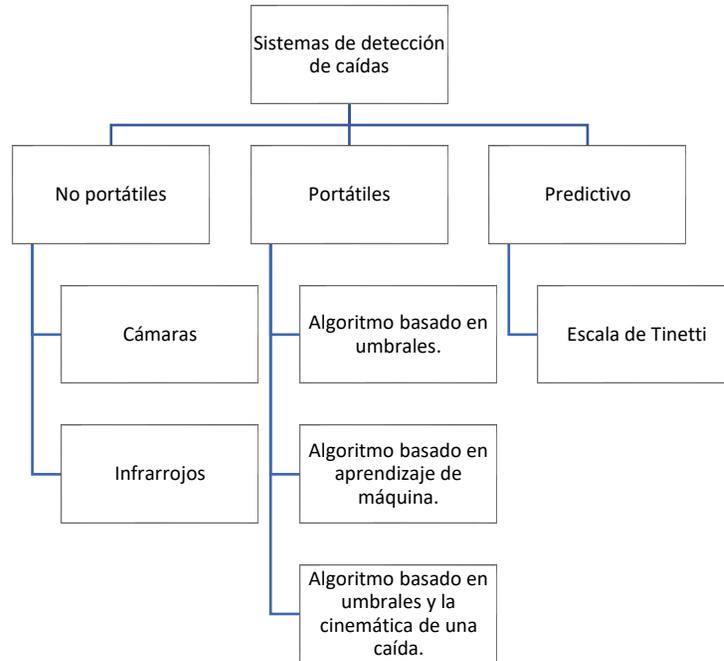


Figura 2.5 Tipos de sistemas detectores de caídas [14], [15].

2.2.2.1 No portátiles.

En esta categoría los sensores se instalan en el entorno en el cual el usuario pasa mayor tiempo, estos sistemas tienen como propósito detectar las perturbaciones que ocurren en su entorno monitorizando su actividad diaria mediante sensores de vibración, proximidad, presión, infrarrojos y cámaras siendo estos dos últimos los más comunes [14].

- **Basado en cámaras**

Este sistema se basa en recopilar imágenes mediante una cámara, en la cual con visión artificial se encarga de identificar la postura humana y con un algoritmo basado en aprendizaje de máquina puede clasificar entre una caída o una actividad normal [16].

Sistema de detección de caídas usando visión artificial.

Este sistema para el reconocimiento de una caída tiene 3 fases, la primera fase consiste en la extracción de la silueta humana del video de entrada. Para diferenciar objetos en movimiento del fondo se usa la resta de fondo (BS). Para este sistema se usó el modelo CodeBook CB como se muestra en la figura 2.6 [16].



Figura 2.6 Resultado de la extracción humana [16].

La segunda fase extrae rasgos particulares y globales a través de la silueta humana como se muestra en la figura 2.7 usando una técnica llamada histograma de proyección que se basa en el uso de un cuadro delimitador que realiza una partición horizontal y vertical [16].

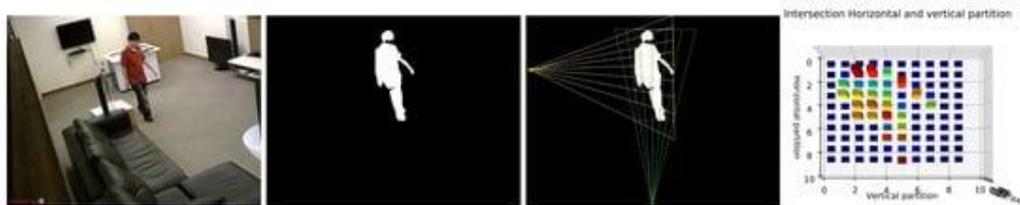


Figura 2.7 Técnica de histograma de proyección [16].

Como tercera fase procesa las imágenes, como se muestra en la figura 2.8 usaron un método de máquina de vector de soporte (SVM) en la cual se encarga de validar si la actividad actual se clasifica como caída o actividad normal. Toma como principales posturas a reconocer agacharse, acostarse, sentarse y pararse ya que son las más parecidas a una caída [16].

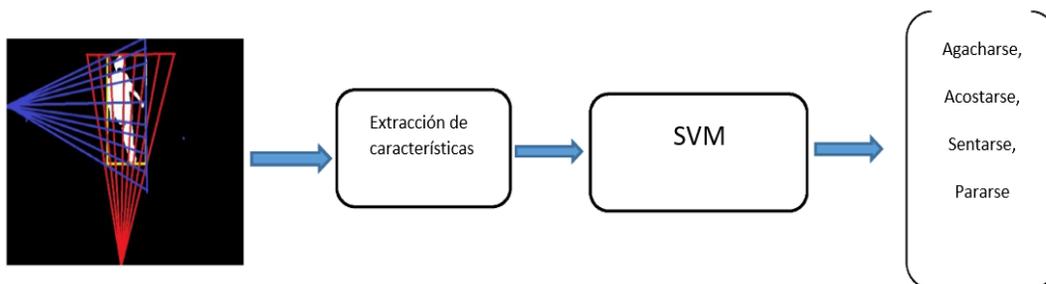


Figura 2.8 Algoritmo de detección de caídas basado en cámaras [16].

Este sistema se implementó en una computadora portátil con un procesador CPU Intel(R) Core (TM) i7-6700HQ y 2,60 Hz y 12,00 GB de RAM logrando una precisión del algoritmo del 97.48%. Este sistema mostro inconvenientes con los falsos positivos con objetos que tenían figura humanoide y no es capaz de identificar dos o más personas al mismo tiempo [16].

- **Basado en infrarrojos.**

Método provisto de un emisor infrarrojo y una cámara de profundidad que permite determinar las actividades que realiza el usuario.

Detección de caídas usando visión artificial y sensor Kinect.

Este sistema utiliza un sensor Kinect de Microsoft, que consta de dos sensores: una cámara RGB y un sensor de profundidad infrarrojo, como se muestra en la figura 2.9 este sistema primero detecta a la persona en la escena usando el método de sustracción de fondo, posteriormente se extrae sus características que describen su postura y finalmente clasifica su acción como una actividad de la vida diaria o como una caída en caso de ser una caída, activa una alarma y hace una llamada de emergencia [17].

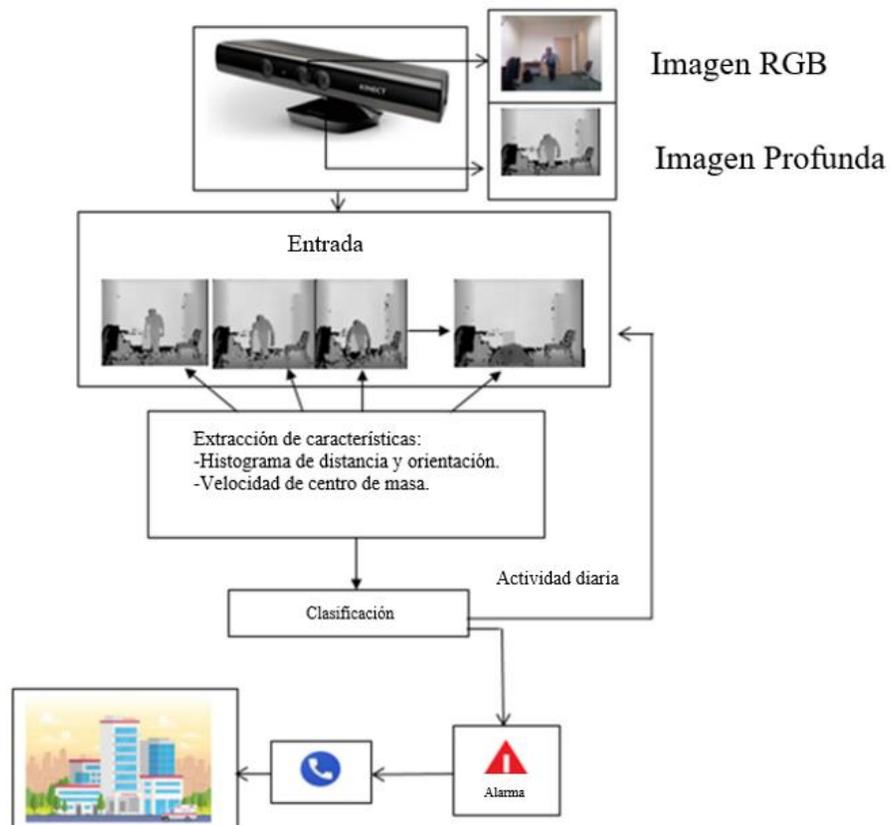


Figura 2.9 Sistema de detección de caídas usando sensor infrarrojo [17].

La precisión obtenida de este algoritmo fue de hasta 98,41% y a diferencia de un sistema de cámaras este al usar el sensor infrarrojo respeta la privacidad de la persona y puede ser usado durante el día y la noche [17].

2.2.2.2 Portátiles

Todos aquellos en los que el usuario usa el dispositivo, este tipo de método usa acelerómetros, sensores de posición e inclinación y algoritmos de control mediante los cuales tiene como objetivo detectar una caída, además de enviar una alarma, también permite implementar dispositivos que amortiguan la caída evitando las fracturas causadas por estos accidentes [14].

Basado en algoritmo de umbrales

- **Sistema de detección y amortiguación de caída usando un algoritmo de umbrales de aceleración y velocidad angular.**

El método utilizado para este sistema fue el uso de un algoritmo de detección de caídas usando la técnica de umbrales como muestra en la figura 2.10 en el caso donde los valores de las señales de aceleración y velocidad angular son mayores a el valor umbral seleccionados de 3m/s^2 y 0.58rad/s respectivamente, la caída será detectada, siendo ese el caso un relé se enciende activando una válvula solenoide que enviará aire comprimido a una bolsa de aire con la finalidad de proteger a la persona de posibles fracturas [18].

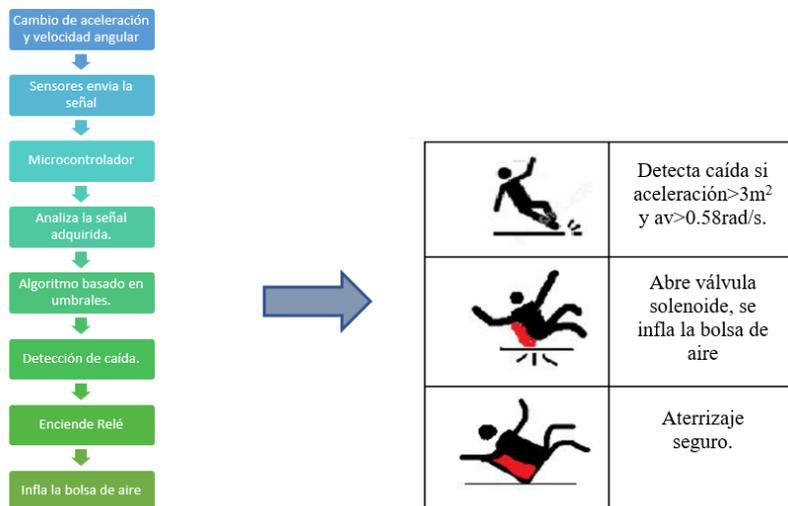


Figura 2.10 Diagrama de flujo que muestra los pasos que sigue este sistema de detección de caídas y el funcionamiento del algoritmo basado en umbrales [18].

- **Sistema de protección y detección de caídas usando un algoritmo de umbrales de velocidad y desplazamiento vertical.**

Este sistema para la detección de caídas usa umbrales de velocidad y desplazamiento vertical dio como resultado 93.6% sensibilidad y especificidad del 95,6%, superior al algoritmo sin desplazamiento con una especificidad del 75,6% [19].

Este algoritmo fue implementado en un sistema embebido, su tiempo de muestreo de 200 Hz en el cual sus características técnicas se detallan en la figura 2.11.

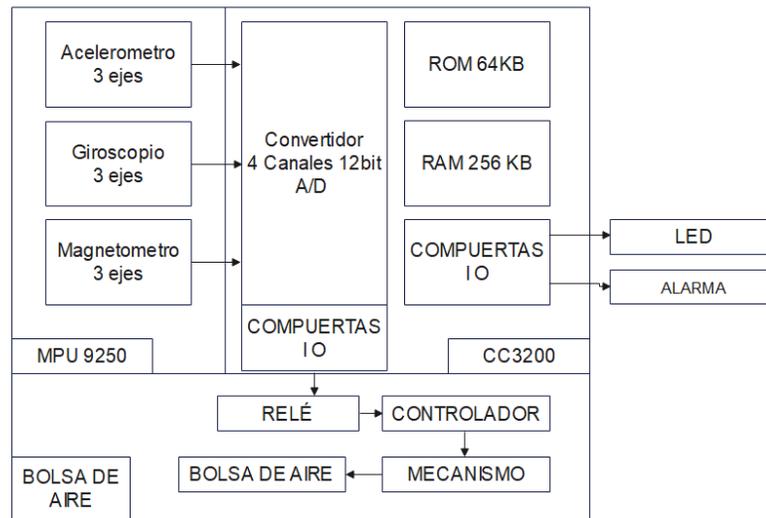


Figura 2.11 Diagrama de bloque de las partes del sistema embebido [19].

En la figura 2.12 muestra un diagrama de flujo en el cual nos detalla el funcionamiento del algoritmo tiene como entrada la aceleración normal y angular en los tres ejes estos datos se los representa en un sistema de coordenadas de aceleraciones del cual se obtiene la aceleración vertical para luego mediante su integración se obtiene la velocidad vertical con el uso de algunos filtros de segundo orden con el fin de eliminar el ruido debido a un ruido derivativo que va en aumento con respecto al tiempo se obtiene la velocidad vertical, con la velocidad previamente filtrada se obtiene el desplazamiento vertical estos dos parámetros de desplazamiento y velocidad vertical son comparados con umbrales seleccionados a partir de un análisis de los datos obtenidos a partir de pruebas experimentales para diferenciar entre una actividad común de una caída (ver figura 2.12) [19].

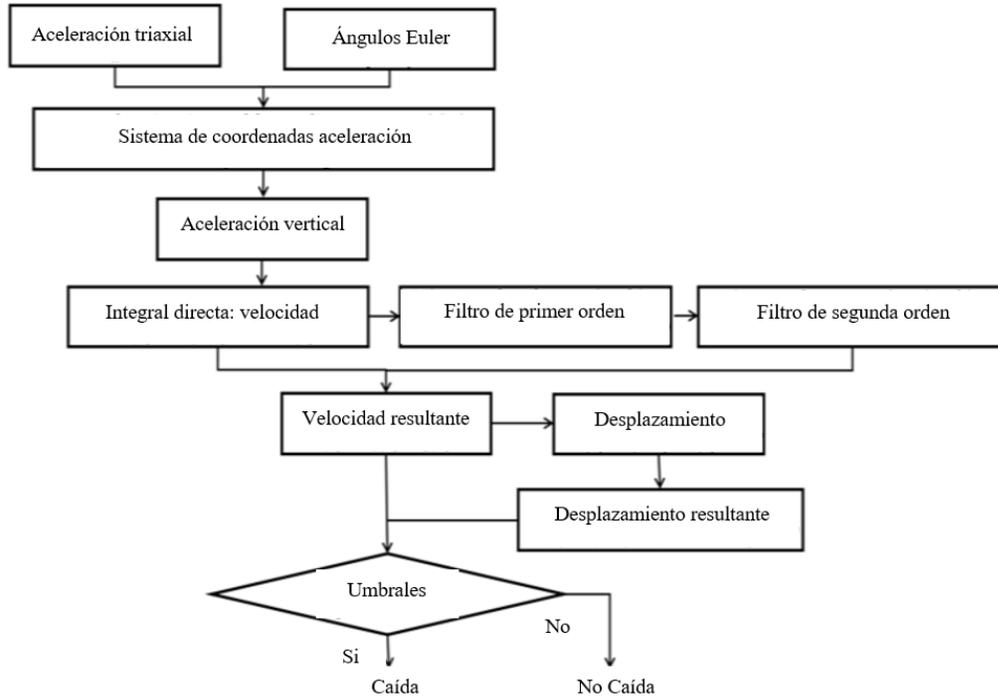


Figura 2.12 Diagrama de flujo del sistema de detección de caídas [19].

Para determinar los umbrales se realiza un análisis de la base de datos que contiene un total de 136 ensayos de actividades de la vida diaria y un total de 188 ensayos de caídas [19], para su fácil entendimiento lo representaron usando un diagrama de caja un modelo visual usado en la estadística descriptiva para visualizar de manera rápida valores máximos, mínimos y mediana de cada una de las actividades como se muestra en la figura 2.13 [20].

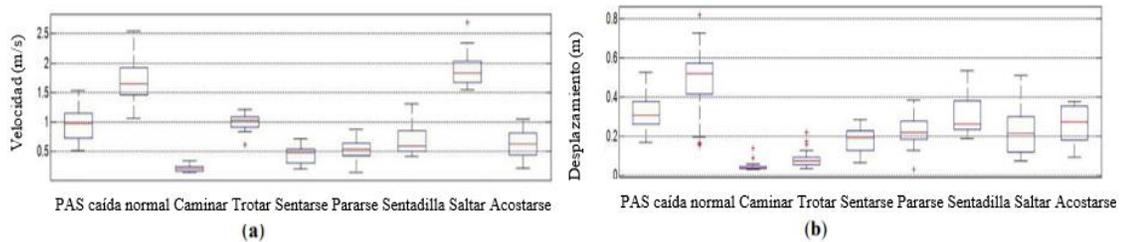


Figura 2.13 (a) Diagrama de cajas de la velocidad máxima y mínima y (b) máximo y mínimo de desplazamiento para cada actividad. Caída parada a sentado (PAS caída) las demás caídas son agrupadas juntas y asignadas a caída normal (Normal caída) [19].

Como se puede observar en la figura 2.13 la mitad de las caídas, de pie a sentado tiene una velocidad vertical máxima inferior de 1m/s, pero este parámetro comparativo, puede detectar también el trotar por lo que se usa también el desplazamiento en el cual después de pruebas, se seleccionaron umbrales de 1 m/s y 0.12 m para la velocidad y desplazamiento respectivamente [20].

- **Algoritmo de detección de caídas usando algoritmo de umbrales crudos del sensor.**

Algoritmo capaz de detectar caídas usando umbrales de valores crudos del sensor en los que se seleccionó valores mínimos y máximos de todas las actividades comunes y caídas, este sistema esta provisto por un acelerómetro y microcontrolador que al detectar una caída enviara una señal de alerta sonora y visual en el dispositivo y un mensaje a la persona responsable del usuario [21]. En la tabla 2.1 indica las partes que componen al sistema además para evitar el ruido del sensor se seleccionó un filtro con una frecuencia de corte entre 0,1 y 0,5 Hz para separar los dos componentes de la orientación estática y el movimiento corporal. [21].

Características del dispositivo

Unidad de detección	Módulo	Rango	Resolución	Precisión
	MMA8451Q	Acelerómetro triaxial	± 8 g	0.0009765625 g/cuentas
Ubicación del dispositivo.	Cintura 0 a 5 cm sobre la cintura.			
Filtro frecuencia de corte	0,1 y 0,5 Hz			

Tabla 2.1. Descripciones de las partes que conforman el sistema embebido.

El algoritmo dio como resultado un 97%, y los resultados se muestran en la Tabla 2.2 esta tabla muestra la probabilidad que el sistema tiene para detectar cada acción, este método funciona bien para personas con un ritmo de actividad bajo a medio [21].

Actividad	Agacharse	Caminar	Acostarse	Sentarse	Subir escaleras	Bajar escaleras	Caerse	Saltar
Agacharse	99							
Caminar	1	97	3					
Acostarse		3	93	4				
Sentarse			4	89	7			
Subir escaleras				7	90	3		
Bajar escaleras					7	95	2	
Caerse						2	97	1
Saltar							1	99

Tabla 2.2 Resultados del algoritmo de detección de caídas [21].

Basado en algoritmo de aprendizaje automático.

- Sistema de detección de caídas usando el algoritmo de aprendizaje de máquina KNN fino.

Este sistema tiene como entrada la aceleración normal y angular en los tres ejes provenientes de sensor de unidad de medición inercial (IMU) el tiempo de muestreo se configuro a 52 Hz y se ubicó en la zona lumbar, se procesan los datos utilizando la plataforma de MATLAB usando el algoritmo KNN fino se entrenó con una base de datos con actividades ya clasificadas y se evaluó con otra base de datos con muestras no clasificadas dando como resultado 99,4% de precisión, en caso de detectar una caída da una señal de alerta [22]. El algoritmo seleccionado fue a partir de pruebas experimentales donde se puso a prueba las opciones de aprendizaje de máquina que ofrece MATLAB siendo el algoritmo de KNN fino el que obtuvo mejores resultados.

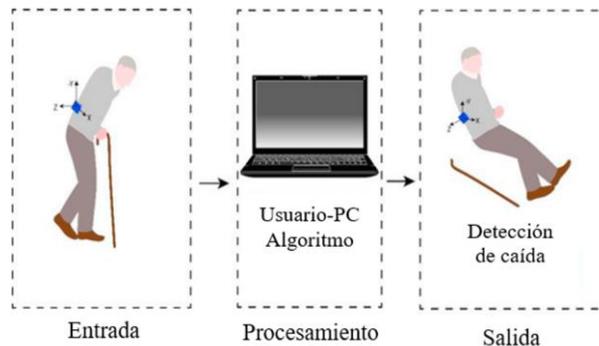


Figura 2.14 Diagrama de bloque del sistema [22].

- **Sistema de detección de caídas usando Fog-Computing, LORA y aprendizaje de máquina.**

Este proyecto muestra una arquitectura que toma los beneficios de varias tecnologías (Fog-Computing, LoRa, Aprendizaje de máquina con el propósito de eliminar las limitaciones que cada uno tiene por separado, en sistemas de monitoreo y control IoT para la salud como el control de caídas, cardiovascular o la diabetes [23].

Para poner a prueba esta arquitectura se toma el caso de uso de detección de caídas, los beneficios otorgados por el uso de este conjunto de tecnologías son disminuir la carga computacional con las compuertas Edge, registro de datos seguro y envío de alertas en tiempo real gracias a la tecnología de la nube con fog- computing y una alta transferencia de datos con un tipo de red inalámbrica de baja potencia y área amplia provisto por LoRa [23].

En la figura 2.15 se muestra un esquema del sistema completo para la detección de caídas, para recopilar datos se usó sensores de unidad de medida inercial MPU9250 estos datos son enviados a las compuertas que ayudan a procesar la señal, determinando el estado en el que se encuentra la persona y enviar dicha información hacia la red LoRa que se encarga de distribuir la información hacia la nube y la base de datos que finalmente el resultado es mostrado en una aplicación web [23].

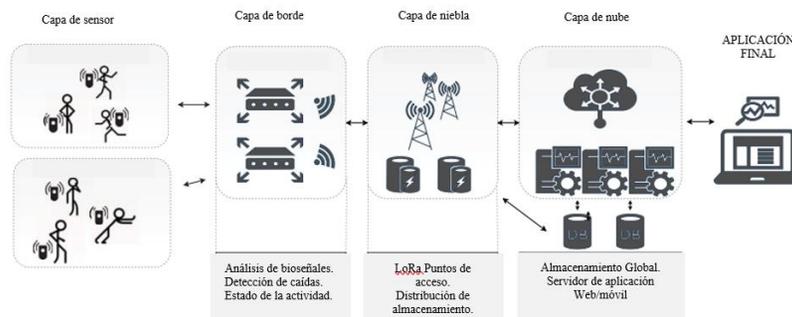


Figura 2.15 Arquitectura propuesta [23].

El algoritmo de aprendizaje profundo se entrenó y evaluó con un conjunto de datos público, se implementó una red neuronal recurrente (RNN) con capas de celdas de memoria a corto plazo (LSTM). La mejor precisión alcanzada fue del $90,10\% \pm 3\%$ [23].

- **Basado en algoritmos que usa la cinemática de una caída y umbrales.**

Sistema portátil de detección de caídas antes del impacto basado en la cinemática de una caída y umbrales

Este sistema fue diseñado con el propósito de reducir todos los puntos negativos del uso de los sistemas de detección portátiles estos parámetros identificados por previos estudios, determinaron que el uso de un algoritmo simple conlleva que las dimensiones, peso y consumo energético de un sistema portátil disminuya [15].

Un algoritmo basado en umbrales si bien es cierto que la precisión de este tipo oscila entre el 80% y 90%, con el uso de modelos matemáticos y características físicas como la altura entre el CG del usuario y el suelo, dio como resultado una especificidad del 97,67%, 94,04% de sensibilidad, precisión del 95.86% y un tiempo de espera de 259 ms un buen tiempo tomando en cuenta que una caída regular toma entre 0,4 y 0,5 segundos [15].

Diseño del algoritmo

Se implementa tres umbrales que se identificaron tomando en cuenta las fases de la caída, como se muestra en la figura 2.18 para empezar si el usuario sufre una pérdida de equilibrio, luego si se encuentra en una caída libre y por último confirmando el impacto para evitar falsas alarmas [15]. Para determinar cada una de las fases tomaron en cuenta modelos matemáticos, ecuaciones y características cinemáticas que describen a una caída.

Para determinar la caída libre y el impacto tiene como parámetro la suma de magnitudes vectoriales (SMV) de la aceleración, se calcula mediante la ecuación 2.1 cada una de sus componentes están expresados en unidades g (9.81 m/s^2) y el valor de cada uno son adquiridos por el sensor de unidad de medida inercial (acelerómetro). Este parámetro es comparado con el umbral seleccionado a partir de los datos adquiridos a través pruebas experimentales que se muestran en la tabla 2.6 para la caída libre el umbral seleccionado es $\pm 0.4g$ con la premisa de que en una caída libre la aceleración tiende a 0 pero tiene que limitar a las actividades de la vida diaria (AVD) para que pueda diferenciar entre una AVD de una caída, se tomó el menor de los valores de una AVD como se observa en la tabla 2.3 el SMV mínimo menor es 0.2 pero no se lo considero ya que en esa actividad no existe suficiente inclinación para que el umbral de fuera de equilibrio pueda superarse y para confirmar el impacto en $\pm 3g$ debido a la aceleración máxima que alcanza el cuerpo en una

AVD, así el sistema puede identificar la caída ya que a partir de $\pm 3g$ los valores SVM máximos son mayores para las caídas [15].

$$|\vec{a}(t)| = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)} \quad (2.1)$$

Actividad	SVM min (g)	SVM Max (g)
Sentarse	0.2	1.8
Pararse	0.4	1.6
Tomar un objeto	0.8	1.2
Caminar	0.5	2.0
Correr	0.35	3.0
Caída hacia atrás	0.09	9.2
Caída hacia adelante	0.085	7.8
Caída hacia la derecha	0.18	6.2
Caída hacia la izquierda	0.13	6.2

Tabla 2.3. Valores mínimos y máximos registrados en actividades de la vida diaria y caídas [15].

Para la determinación de pérdida de balance se usaron modelos matemáticos para describir la trayectoria para ello se utiliza el ángulo de inclinación del eje y que se obtiene a partir de la ecuación 2.2 [15].

$$\beta = \cos^{-1} \left(\frac{a_y(t)}{\sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)}} \right) \quad (2.2)$$

El algoritmo se elaboró a partir de la Ecuación de Torricelli cinemática clásica ecuación. 2.3 que describe un movimiento lineal con aceleración constante a lo largo de un eje [15].

$$v^2 = v_0^2 \pm 2a\Delta x \quad (2.3)$$

Donde v , v_0 representan el inicio y final de las velocidades en (m/s) del cuerpo, Δx es la distancia recorrida en metros y a es la aceleración en (m/s²) [15].

Considerando que la caída tiene un movimiento angular, durante una caída estática, la persona realiza un movimiento de rotación alrededor el eje que se puede aproximar entre el punto de contacto con el suelo (los pies) y el centro de masa. La Ecuación de Torricelli Ecuación 2.4 también se puede presentar para describir un movimiento cinemático de rotación [15]:

$$w^2 = w_0^2 \pm 2a\Delta\theta \quad (2.4)$$

Donde w , w_0 representan el inicio y final de las velocidades angulares durante una caída en (rad/s), $\Delta\theta$ es el ángulo de inclinación expresada en radianes y a es la aceleración angular expresada en (rad/s²) [15].

El modelo matemático que define una caída lateral está dado por la ecuación 2.5.

$$w^2 = \frac{3g(1 - \cos\varphi)}{l} \quad (2.5)$$

Donde l es la distancia entre la cintura y el suelo en metros, φ es el ángulo del eje cuando cae en radianes g es la gravedad en (m/s²) [15].

Además Mastorakis, aproximó la caída de una persona como una varilla cayendo con distribución de masa uniforme. Para este modelo se consideró que la única aceleración sobre el movimiento es la gravedad g . El modelo indica que la velocidad es proporcional a la altura, una persona más alta tarda más en caer, y en el momento del impacto, tiene una mayor velocidad. [24] Sin embargo, la aceleración es independiente de la altura. Con este concepto se definió la ecuación 2.6 [15].

$$w_n = \sqrt{w_{n-1}^2 + \frac{3g(\cos\beta_{n-1} - \cos\beta_n)}{h_1}} \quad (2.6)$$

Donde h_1 es la distancia en centímetros entre la PCB y el suelo.

La ecuación 2.5 y 2.6 demostraron que el desplazamiento angular Δs (cm) de una persona entre dos adquisiciones $n-1$ y n , puede ser definida por la ecuación 2.7 [15].

$$\Delta s = \frac{3}{2} h_1 x [\cos(\beta_{n-1}) - \cos(\beta_n)] \quad (2.7)$$

Para obtener el desplazamiento total en un intervalo de tiempo t se describe la ecuación 2.8 [15].

$$S = \sum \Delta s = \sum_1^n \frac{3}{2} h_1 x [\cos(\beta_{n-1}) - \cos(\beta_n)] \quad (2.8)$$

La tercera característica importante es la distancia normal d_n , se asume que h_1 es igual a d cuando se está estático [15].

$$d_n = h_1 \cos\beta_n \quad (2.9)$$

Para determinar la pérdida de balance se determinó que $d - S \leq h_2$ y Δh_2 se seleccionó mediante pruebas experimentales en la cual para mejorar la robustez del algoritmo se seleccionó un umbral de 1.2. Todas estas ecuaciones y fases para mejorar su comprensión se muestran en la figura 2.16. y figura 2.17 [15].



Figura 2.16 Diagrama de flujo de las fases y características cinemáticas de una caída [15].

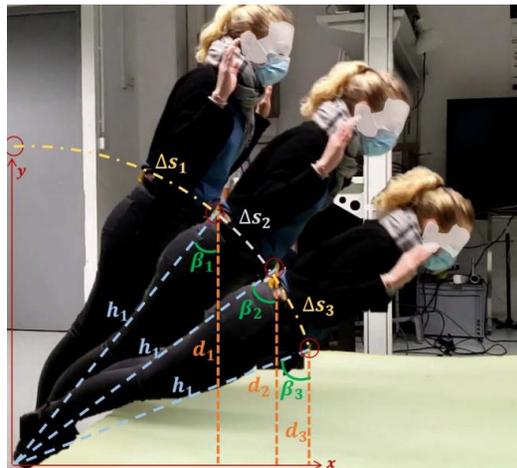


Figura 2.17. Simulación de una caída hacia adelante mostrando cada uno de los parámetros previamente descritos en las ecuaciones [15].

Para las pruebas y registro de datos para los experimentos realizados para su validación se usó la configuración y componentes que se detallan en la tabla 2.6. logrando que el sistema tenga un bajo consumo energético de 5,91 mA [15].

		Rango	Resolución	Estabilidad
Unidad de detección	LIS3DH	Acelerómetro ± 8 g	12 bits	0.0001 g
		triaxial		
Ubicación de la PCB.	Cintura 0 a 5 cm sobre la cresta iliaca usando un cinturón.			
Bluetooth	Bluetooth de baja energía. (BLE)			
Microcontrolador	nRF52832			
Muestreo	400 Hz			
Dirección de los ejes del sensor	Orientado con el eje x para la izquierda; el eje y hacia abajo positivo y el eje z hacia adelante positivo.			

Tabla 2.4 Descripción de la configuración y partes que conforman el sistema embebido del sistema.

- Evaluación del desempeño

La fiabilidad del sistema se mide según los resultados que se interpretan según los siguientes parámetros estadísticos [15]:

- La sensibilidad (SE) es la capacidad del sistema para identificar caídas [15].

$$SE = \frac{VP}{VP + FN} \quad (2.10)$$

- La especificidad (ES) es la capacidad de detectar correctamente AVD como AVD [15].

$$ES = \frac{VN}{VN + FP} \quad (2.11)$$

- La precisión (PR) es la capacidad del sistema para discernir, caída de una AVD [15].

$$PR = \frac{SE + ES}{2} \quad (2.12)$$

Donde VP son los verdaderos positivos, VN son los verdaderos negativos, FP son los falsos positivos y FN son los falsos negativos [15].

Tiempo de espera: es el tiempo entre la detección de la caída hasta el impacto [15].

2.2.2.3 Predictivo

Escala de Tinetti

Esta escala ayuda a detectar el riesgo de caídas en ancianos en un periodo de un año. Realiza una aproximación realizando una serie de preguntas de opción múltiple que se divide en dos partes: Parte I. Equilibrio y Parte II. Marcha en la cual cada pregunta es valorada y se obtiene una puntuación total en la cual se considera riesgo alto de caídas: menos de 19 puntos; riesgo de caídas: de 19 a 23 puntos, y riesgo bajo o leve: de 24 a 28 puntos [4].

2.2.3 Sistema de adquisición de datos: Acondicionamiento de señal.

Para el estudio de un sistema físico es necesario obtener datos de su funcionamiento lo que normalmente se realiza a través de sensores [25], estos datos necesitan ser filtrados debido a que la IMU se trata de un dispositivo real, lo que provoca que cualquier perturbación en el entorno produzca que se mezcle una señal no deseada con la señal de nuestro interés [26].

2.2.3.1 Tipos de filtros

El filtrado se refiere al proceso de eliminar ciertas bandas de frecuencia para evitar que se recepan señales que no son de interés y deja pasar otras bandas de frecuencia, existen dos bandas la de paso que es el rango de frecuencias que pasa por el filtro y la banda de supresión que son el rango de frecuencias que se rechaza, el límite que permite que bandas de frecuencia pasan o se rechazan se lo denomina como frecuencia de corte [27]. Los filtros se clasifican de acuerdo con los rangos de frecuencias que transmiten o rechazan:

Filtro pasa bajas

Permite el paso a las señales con una frecuencia menor a la frecuencia de corte este filtro se puede diseñar a partir de la ecuación 2.13 [25].

$$\omega_c = \frac{1}{RC} = \frac{1}{\tau} \quad (2.13)$$

Donde:

ω_c es la frecuencia de corte.

τ es la constante de tiempo.

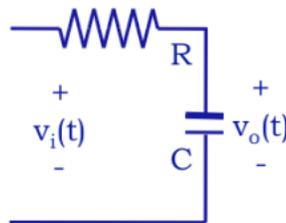


Figura 2.18 Circuito filtro pasa bajo RC serie [25].

Filtro pasa altas

Permite el paso a las señales con una frecuencia mayor a la frecuencia de corte [25] comparten la misma ecuación que el filtro pasa bajo.

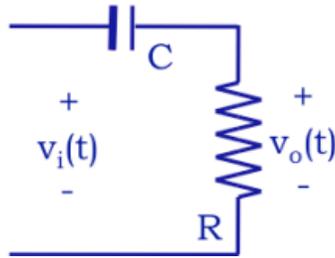


Figura 2.19 Circuito filtro pasa alto RC serie [25].

Filtro digital pasa bajo de media móvil.

Utilizados a menudo para limpiar señales, eliminar ruido, crear un efecto de suavizado, calcular los promedios de datos, y diseñar diezmadore e interpoladores que son implementados mediante la siguiente ecuación [27].

$$A_n = \alpha * M + (1 - \alpha) * A_{n-1} \quad (2.14)$$

Donde A_n es la señal filtrada, α coeficiente con el cual el filtro elimina el ruido se recomienda un valor de 0.05, M es la señal sin filtrar y A_{n-1} es la señal anterior a A_n que normalmente se lo inicializa en 0.

2.2.4 Protocolo simple de transferencia de correo.

Protocolo de red usado para el envío de correos electrónicos entre dispositivos se basa en el modelo cliente – servidor donde un dispositivo se encargar de enviar el correo a uno o más dispositivos [28].

CAPÍTULO III

3 Marco metodológico

3.1 Modelo de la Investigación

En el presente trabajo se desarrolló una investigación aplicada, ya que está orientada en resolver un problema [29]. En este caso se trata de reducir o evitar los daños causados por una caída, mediante el diseño de un algoritmo que sea capaz de identificar una caída de una actividad común con el propósito de que se implemente en dispositivos electrónicos portátiles.

Además, se usa investigación documental ya que se recopila y selecciona información a través de la lectura de documentos como artículos científicos y libros [30]. También se usa la investigación descriptiva para especificar los detalles del funcionamiento del algoritmo y para concluir se aplica una investigación experimental ya que se implementa el algoritmo en un dispositivo portátil para realizar simulaciones de caídas sobre un ambiente controlado con el fin de determinar la precisión de este, identificar fallos para realizar correcciones.

3.2 Diseño de la Investigación

Para el cumplimiento de los objetivos, se plantean las siguientes actividades con la finalidad de seguir de forma secuencial el proceso que se desarrolló, para el diseño y evaluación de desempeño del algoritmo.

3.2.1 Investigación acerca de los algoritmos de detección de caídas.

En esta fase se usa la investigación documental en la cual se recopila información sobre los diferentes tipos de detectores de caídas para personas de la tercera edad existentes con el propósito de obtener criterios que ayuden a diseñar el algoritmo.

Actividad 1 “búsqueda de documentos científicos acerca de los métodos existentes para detectar una caída en personas de la tercera edad”; en esta actividad se obtiene información acerca de los métodos usados para detectar una caída en personas de la tercera edad, con un enfoque en los dispositivos portátiles.

Actividad 2 “análisis de los métodos de detección de caídas para personas de la tercera de edad”; en esta etapa se seleccionaron los métodos más significantes a partir de un grupo de documentos científicos.

3.2.2 Diseño de un algoritmo de detección de caídas.

Esta etapa se realizará, basándose en la metodología descriptiva el diseño del algoritmo de detección de caídas para personas de la tercera edad.

Actividad 1 “determinación de los requerimientos del algoritmo”; se establecen las características que debe cumplir el algoritmo.

Actividad 2 “selección del método de detección de caída”; en esta actividad se seleccionará el método más eficiente basando en las ventajas y desventajas de cada tipo portátil, no portátil o predictivo.

Actividad 3 “selección del tipo de algoritmo”; en base a las características que debe cumplir se selecciona el tipo de algoritmo que se va a usar.

Actividad 4 “se plantea alternativas para su solución”; se plantea opciones que solucionan el problema.

Actividad 5 “selección de la mejor alternativa”; en base a comparación de los criterios, se determina la solución que más beneficios brinde.

3.2.3 Evaluación del funcionamiento del algoritmo.

En esta fase, se usa la investigación descriptiva para especificar las características del sistema mediante planos y esquemas eléctricos, además se usa la metodología experimental en la cual se realizan simulaciones de movimientos comunes y caídas en un ambiente controlado con equipos de protección para determinar la precisión del algoritmo.

Actividad 1 “selección de sensor, filtro, materiales y controlador”; la selección se basa en la disponibilidad del mercado y características técnicas.

Actividad 2 “elaboración de esquemas de conexiones”; en esta etapa se elaboran los esquemas de conexión para la construcción del dispositivo donde se evaluará el algoritmo.

Actividad 3 “construcción de un mecanismo mecánico para validar el funcionamiento del sensor”; se diseña y construye un mecanismo que simula un movimiento circular, con soporte para el sensor para comprobar la programación en el microcontrolador.

Actividad 4 “Programar el modelo matemático de una caída en código C++ y cargar el programa al microcontrolador”; esta etapa se realiza para validar las unidades de medida, resultados de las ecuaciones y filtros.

Actividad 5 “Programar el algoritmo completo en el microcontrolador”; en esta actividad se implementa el algoritmo completo con umbrales y actuadores

Actividad 6 “Diseño y construcción del sistema de sujeción para el cuerpo y el sensor”; se realiza un diseño y construcción donde se instala el sensor, controlador y demás componentes para realizar las pruebas experimentales.

Actividad 7 “Pruebas experimentales”; en un ambiente controlado en una muestra de personas jóvenes se realizarán pruebas de diferentes tipos de caídas, en una muestra de personas de la tercera edad se realizan las actividades comunes, los resultados de las actividades se registran en una matriz de confusión.

Actividad 8 “Ajustes y correcciones”; en esta etapa se corregirá los errores del algoritmo.

Actividad 9 “Evaluación de desempeño”; de los resultados obtenidos se obtendrá los parámetros de sensibilidad, especificidad y precisión.

CAPÍTULO IV

4 Resultados y análisis

En este capítulo se seleccionará el tipo de detector de caída, va a ser orientado al tipo de algoritmo y se describirá las especificaciones de los requerimientos necesarios para el diseño del mismo. Posteriormente se presentarán alternativas y se seleccionara el apropiado en base a las especificaciones identificadas.

Para evaluar el desempeño del algoritmo se diseñará un sistema embebido portátil para evaluar su funcionamiento mediante una matriz de confusión.

4.1 Especificaciones del algoritmo a diseñar.

En esta etapa se describirán los parámetros y restricciones que debe cumplir el algoritmo.

Dirigido a. personas mayores a 65 años.

Detecta varios tipos de caídas. el algoritmo sea capaz de determinar la mayor cantidad de tipos de caídas como caídas con o sin flexionar rodillas.

Implementar en dispositivos portátiles. el algoritmo se pueda implementar en un dispositivo de bajo consumo energético y portátil.

Alta precisión. el sistema debe tener un alto porcentaje de acierto en diferenciar una caída de una AVD.

Algoritmo de bajos requerimientos de hardware: el uso de operaciones simples para detectar caídas permite que pueda ser implementado en dispositivos portátiles de menor tamaño, peso, consumo energético y costo.

Materiales. los materiales seleccionados puedan ser adquiridos en el mercado local y que cumplan con los requerimientos técnicos necesarios.

Posea un tiempo de espera. detecte la caída antes del impacto.

4.2 Selección del método de detección de caída.

Para seleccionar el método de detección de caída se realizó un cuadro comparativo con el propósito de identificar las ventajas y desventajas de cada tipo.

Tipos de detector de caída	Ventajas	Desventajas
No portátiles.	<ul style="list-style-type: none"> • Utiliza una corriente eléctrica estable. • Monitorización constante. • Esta instalado en la vivienda del usuario. • Alto porcentaje de acierto en ciertos tipos de caídas. • Permite que el propio paciente y el sistema avise a algún familiar o a emergencias. • Permite que el paciente desactive la alarma en caso de falsos positivos. • Ubica al usuario de manera exacta el lugar del accidente. 	<ul style="list-style-type: none"> • Necesita varios sensores para evitar puntos ciegos. • Mayor gasto energético. • Riesgo para la privacidad del usuario. • Solo funciona dentro del área designada. • Los objetos del entorno influyen en la detección de la caída • Costo elevado.
Portátiles	<ul style="list-style-type: none"> • Alto porcentaje de acierto en ciertos tipos de caídas. • Utiliza un único dispositivo. 	<ul style="list-style-type: none"> • Es necesario el uso de una fuente de alimentación: batería, pilas. • El usuario tiene que colocarse el dispositivo.

	<ul style="list-style-type: none"> • Monitorización constante. • Menor costo que los de monitorización del entorno. • Menor consumo energético. • Los obstáculos del entorno no influyen en la detección de caída. • Envío automático de alerta en caso de caída el usuario puede cancelar en caso de falso positivo. • Se puede instalar un sistema de amortiguación para evitar que el usuario sufra daños a causa de la caída. 	<ul style="list-style-type: none"> • Probabilidad que el dispositivo sufra daños cuando se produce una caída.
Predictivo	<ul style="list-style-type: none"> • Detecta de manera anticipada si va a tener una caída en un lapso de un año. • Alta fiabilidad 	<ul style="list-style-type: none"> • Un rango muy amplio en el cual va a suceder una caída. • No hay una acción para evitar el problema.

Tabla 4.1 Ventajas y desventajas de los sistemas de caída portátiles, no portátiles y predictivos [18], [19], [21], [22], [23], [15].

Para este trabajo se seleccionará un algoritmo que pueda ser implementado en sistemas portátiles, como se observa en la tabla 4.1 existen más ventajas en el uso de un sistema portátil a diferencia de los otros sistemas en el cual el más importante es la posibilidad de implementar sistemas de

amortiguamiento de impacto causado por una caída evitando las consecuencias físicas, económicas y psicológicas que conllevan este accidente.

4.3 Selección de tipo de algoritmo

Para la selección del tipo de algoritmo a usar, se identificará de forma general las características de cada uno de ellos basados en las especificaciones previamente seleccionadas.

Tipo de Algoritmo	Características
Umbral Sistema de detección y amortiguación de caída usando un algoritmo de umbrales de aceleración y velocidad angular	<ul style="list-style-type: none"> • Tipos de caídas que detecta desconocido. • Precisión desconocida. • Algoritmo simple. • Sensores usados para la adquisición de datos no son comerciables en la localidad. • Tiempo de espera desconocida. • Filtro desconocido.
Umbral Sistema de protección y detección de caídas usando un algoritmo de umbrales de velocidad y desplazamiento vertical.	<ul style="list-style-type: none"> • Detecta dos tipos de caídas. • Precisión alta de 94,6 % • Algoritmo complejo. • Sensor MPU 9250 hay disponibilidad en el mercado. • Tiempo de espera 363 ± 174ms. • Filtro de segundo orden.
Umbral Algoritmo de detección de caídas usando algoritmo de umbrales crudos del sensor	<ul style="list-style-type: none"> • Detecta solo el tipo de caídas sin flexionar las rodillas. • Precisión del 97%. • Algoritmo complejo ya que detecta cada tipo de actividad y caída.

		<ul style="list-style-type: none"> • Sensor usado no es de fácil adquisición. • Tiempo de espera desconocido solo es usado para avisar a sus cuidadores. • Filtro de primer orden.
Aprendizaje automático	Sistema de detección de caídas usando el algoritmo de aprendizaje de máquina KNN fino.	<ul style="list-style-type: none"> • Tipos de caídas que detecta desconocido usaron base de datos públicas para entrenar y comprobar su funcionamiento. • Precisión alta de 99.4%. • Algoritmo complejo. • Sensor desconocido. • Tiempo de espera desconocido. • Filtro desconocido.
	Sistema de detección de caídas usando Fog-Computing, LORA y aprendizaje de máquina.	<ul style="list-style-type: none"> • Tipos de caídas que detecta desconocido se enfocó en la arquitectura que usa. • Precisión del 90,1%. • Algoritmo complejo. • Sensor MPU 9050 existe en el mercado local. • Tiempo de espera desconocido. • Filtro desconocido.
Cinemática de una caída y Umbrales	Sistema portátil de detección de caídas antes del impacto basado en umbrales de aceleración y desplazamientos.	<ul style="list-style-type: none"> • Tipos de caídas detecta con precisión las caídas sin flexionar rodillas. • Precisión alta 97,7%. • Algoritmo de bajo requerimientos. • Sensor LIS3DH no hay disponibilidad en el mercado local. • Tiempo de espera de 259 ms. • Filtro desconocido.

Tabla 4.2 Identificación de las características de cada tipo de algoritmo [18], [19], [21],

[22], [23], [15].

Como se observa en la tabla 4.2, el tipo de algoritmo óptimo a seleccionar sería el algoritmo basado en la cinemática de una caída y umbrales, debido a su alta precisión, bajos requisitos para su implementación en hardware y tiempo de espera.

4.4 Metodología para el desarrollo del algoritmo.

El algoritmo de detección de caídas estará basado en la cinemática de una caída y umbrales. En trabajos previos se realizó este tipo de algoritmo, pero como muestra la figura 2.17 ellos evaluaron su algoritmo con tipos de caída sin flexionar las rodillas, en su trabajo mencionan que flexionar las rodillas y colocar las manos son reacción propia de las personas para protegerse, como se muestra en la figura 4.1 donde se registraron caídas de personas en centros de larga instancia, donde se observa que durante este accidente se tiende a flexionar las rodillas, por ende se propone un algoritmo que sea capaz de detectar caídas con o sin flexionar las rodillas.

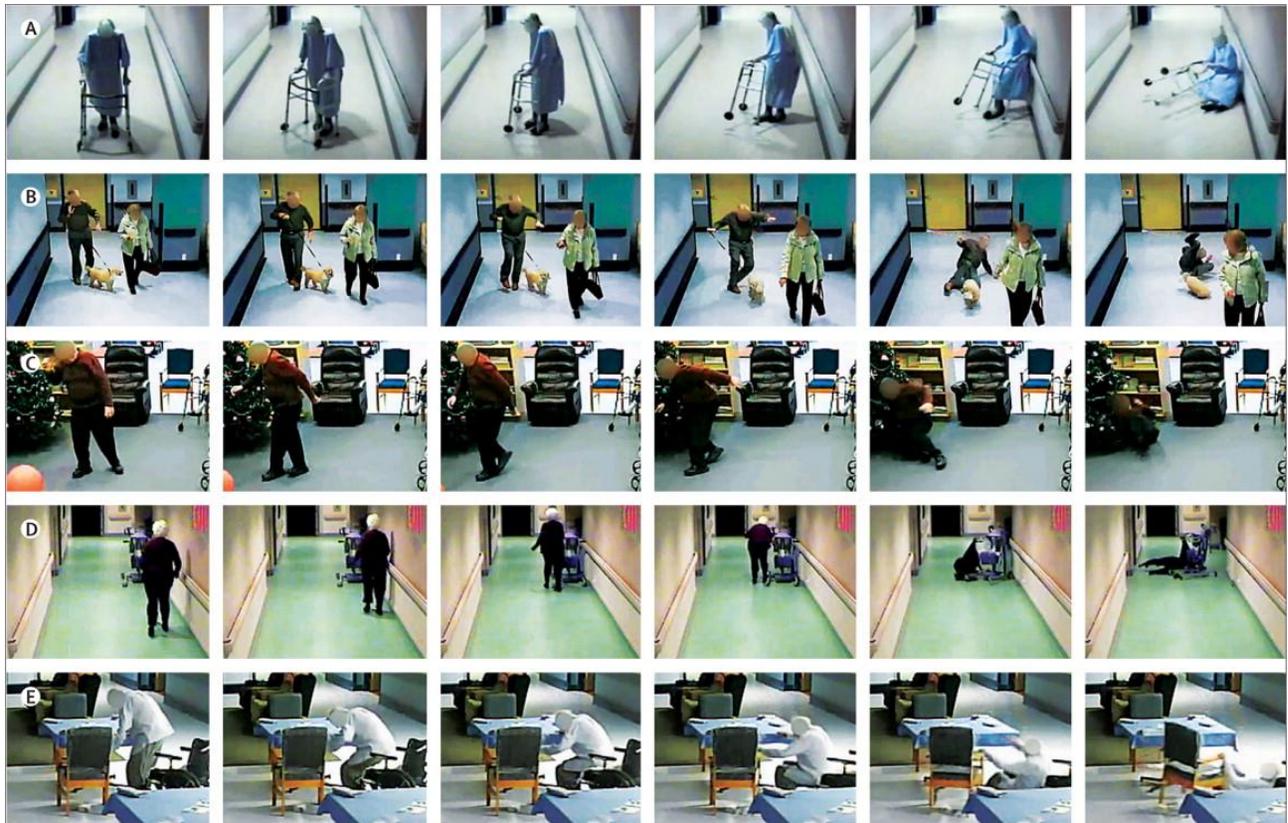


Figura 4.1 Fotogramas de una cámara de video de accidentes de caída en personas de la tercera edad [31].

4.4.1 Selección de componente para el sistema embebido para la validación del algoritmo.

Para comprobar el funcionamiento del algoritmo se implementó sobre un sistema embebido del cual se seleccionaron sus componentes tomando en cuenta su disponibilidad en el mercado local y características técnicas.

4.4.1.1 Selección de sensor:

En base a sus dimensiones, precio y sus características se utilizará un sensor del modelo MPU su versión 6050 mostrado en la figura 4.2 ya que posee 6 grados de libertad con 3 ejes para el acelerómetro y 3 ejes para giroscopio. Además de sus características que se especifican en la tabla 4.3.

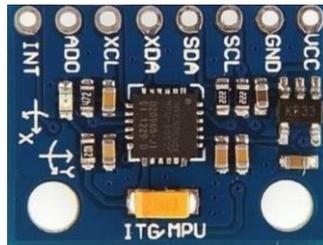


Figura 4.2 Sensor MPU 6050 [32].

Característica	Descripción
Rango Acelerómetro	2g/4g/8g/16g
Rango de Giroscopio	250Grad/Seg, 500Grad/Seg, 1000Grad/Seg, 2000Grad/S eg
Convertor análogo digital	16 Bits
Protocolos de comunicación	I2C
Dimensiones	20 mm x 16 mm x 03 mm
Voltaje de alimentación	3,3 v

Tabla 4.3 Especificaciones del sensor MPU – 6050 [32].

4.4.1.2 Selección de filtro:

Debido a que se va a usar sobre un dispositivo portátil se seleccionó un filtro digital al ángulo obtenido a partir del sensor ya de este parámetro se basa las demás ecuaciones, para reducir tamaño

y peso, el mismo es representado con la ecuación 2.14 pasa bajo para el acelerómetro debido a su sensibilidad que genera ruido y pasa alto para el giroscopio debido a un error derivativo.

4.4.1.3 Selección de microcontrolador:

Se selecciona dos versiones de microcontroladores del modelo ESP el 8266 mini D1 y WROOM-32 que se muestran en la figura 4.3 debido a sus características, precio, dimensiones y la conexión inalámbrica que ofrece ya que se piensan realizar pruebas simuladas de una caída se trata de limitar el uso de cables.

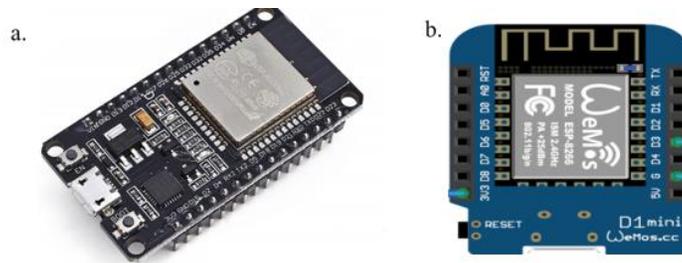


Figura 4.3 a) Microcontrolador ESP-32 WROOM [33]. b) Microcontrolador ESP el 8266 mini D1 [34].

Entre las características técnicas por la cual se seleccionó estos microcontroladores se muestra en la tabla 4.4.

Placa de Desarrollo	Especificaciones Técnicas	Unidades
ESP – WROOM 32	CPU	Dos núcleos de 32 bit
	Frecuencia de reloj	240MHz
	Voltaje de alimentación y de entradas y salidas.	3,3 v
	Dispone de un regulador de voltaje.	Vmax 12 v Vmin 3,3 v
	SRAM	520KB
	Protocolo de conexión	I2C, ESP NOW
	WIFI	2.4 ~ 2.5 GHz
	Peso	12 gramos
Dimensiones	50mmx29mmx12mm	
ESP 8266 WEMOS D1 MINI	CPU	Un núcleo de 32 bit
	Voltaje de alimentación y de entradas y salidas.	3,3 v
	Frecuencia de reloj	80MHz/160MHz
	Dispone de un regulador de voltaje.	Vmax 12 v Vmin 3,3 v
	RAM	96KB
	Protocolo de conexión	I2C, ESP NOW

WIFI	2.4 ~ 2.5 GHz
Peso	6 gramos
Dimensiones	35mmx26mmx12mm

Tabla 4.4 Especificaciones del microcontrolador ESP – 32 y ESP 8266 [33],[34].

4.4.1.4 Selección de baterías

Se utiliza batería li-lion de la marca ultraFire de 3.7v debido a su voltaje que está dentro del rango permitido del regulador de voltaje que tienen incorporados los microcontroladores, y 6800 mAh que permitirá la mayor autonomía del dispositivo para realizar las pruebas (ver figura 4.4).



Figura 4.4 Batería de li-on.

4.4.2 Diseño del sistema embebido para la adquisición de datos y validación del algoritmo.

El esquema de conexiones del circuito se muestra en la figura 4.5 está compuesto por el microcontrolador ESP-WROOM 32 que se encarga de recibir, procesar, filtrar los datos del sensor, con el propósito de determinar la caída y enviar un correo electrónico de alerta, energizado por una batería de 3.7 v, con dos leds indicadores el led azul que indica que se ha detectado una caída y el envío del correo electrónico, el led rojo indica que el pulsador esta accionado que se encarga de apagar la alarma.

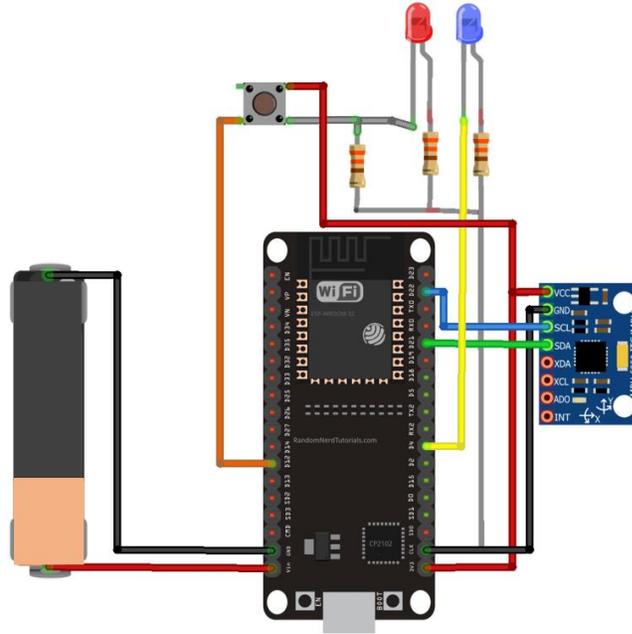


Figura 4.5 Esquema de las conexiones del dispositivo uno.

4.4.3 Diseño del algoritmo.

Para el diseño del algoritmo se tomó en cuenta las fases de una caída las cuales son el desequilibrio, caída libre e impacto y su relación que tienen estas con la magnitud de la aceleración total y la cinemática de una caída.

Este sistema está basado en el algoritmo mostrado en la figura 2.16 con las siguientes diferencias al usar un sensor con acelerómetro y giroscopio se toma también la aceleración angular en “y” usando la ecuación 4.1 [35], para obtener el ángulo de inclinación con el giroscopio e implementar el filtro digital de media móvil también conocido como filtro complementario además también se usó este filtro para la aceleración total, se configura el dispositivo como receptor y por último se implementó un servidor smtp para enviar un correo de alarma a la persona que se encarga del cuidado del usuario.

$$\beta_g = \left(\beta_{g(n-1)} + \frac{g_y}{131} \right) \cdot dt \quad 4.1$$

Antes de su implementación se comprobó el funcionamiento de cada una de las ecuaciones para facilitar su estudio se realizó un soporte para el sensor mostrado en la figura 4.6 simula el movimiento de una caída sin flexionar las rodillas.



Figura 4.6 Soporte para el sensor MPU – 6050.

Para validar el funcionamiento del filtro digital se usó el graficador de Arduino el resultado se muestra en la figura 4.7 se puede observar el resultado del uso de filtro digital en la cual la señal azul indica el ángulo de inclinación sin filtro y la línea roja el ángulo de inclinación usando el filtro.

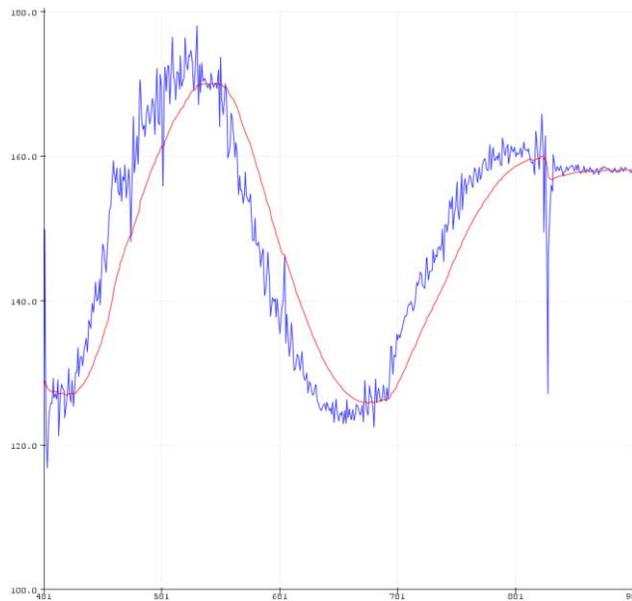


Figura 4.7 Señal filtrada del sensor MPU-6050 usando el filtro complementario de media móvil.

El soporte ya mencionado ayudo también a verificar los resultados obtenidos de las ecuaciones 2.2, 2.7 y 2.8 con ayuda también de pruebas de escritorio realizadas en una hoja de cálculo para verificar los resultados esta prueba se puede observar dentro del Anexo A, la tabla se coteja con

los resultados obtenidos del monitor serial de Arduino esta prueba se realizó con el fin de verificar los valores y en las unidades en las que se está trabajando.

Con estos preparativos se presenta el algoritmo que se muestra en un diagrama de flujo en la figura 4.8. Donde en el lado izquierdo indica la descripción de cada fase del algoritmo y en la parte derecha indica los datos y operaciones que se realiza para comprobar cada fase.

Este algoritmo se divide en tres fases la cual la primera fase inicializa el sistema en esta parte se determinan las constantes del sistema en este caso son las características físicas de la persona y que el sensor este operando determinando la aceleración en cada eje, velocidad angular en el eje y, tiempo de muestreo, tiempo del desplazamiento y umbral.

En la segunda fase se describe el algoritmo el cual se encarga de identificar una caída, en donde se extraen sus características cinemáticas, aceleración y ángulo de inclinación después se filtran usando el filtro de media móvil para después calcular la distancia normal y el desplazamiento sobre un periodo de tiempo.

Para determinar la caída se tiene que cumplir dos condicionales primera si existe la perdida de equilibrio que se determina mediante la diferencia de la distancia normal y el desplazamiento total si esta diferencia es menor o igual al umbral se detecta una caída la segunda condición sucede cuando el cuerpo está en caída libre esto se determina con la aceleración mínima registrada en cada actividad.

Tercera fase en caso de dar falso a una de las condicionales previas el algoritmo se queda en esa fase y en caso de dar positivo a las condicionales pasa a la fase de alarma que se encarga de dar una señal de alerta que se traduce como una señal eléctrica que encendería un indicador visual de color azul y envió del correo electrónico.

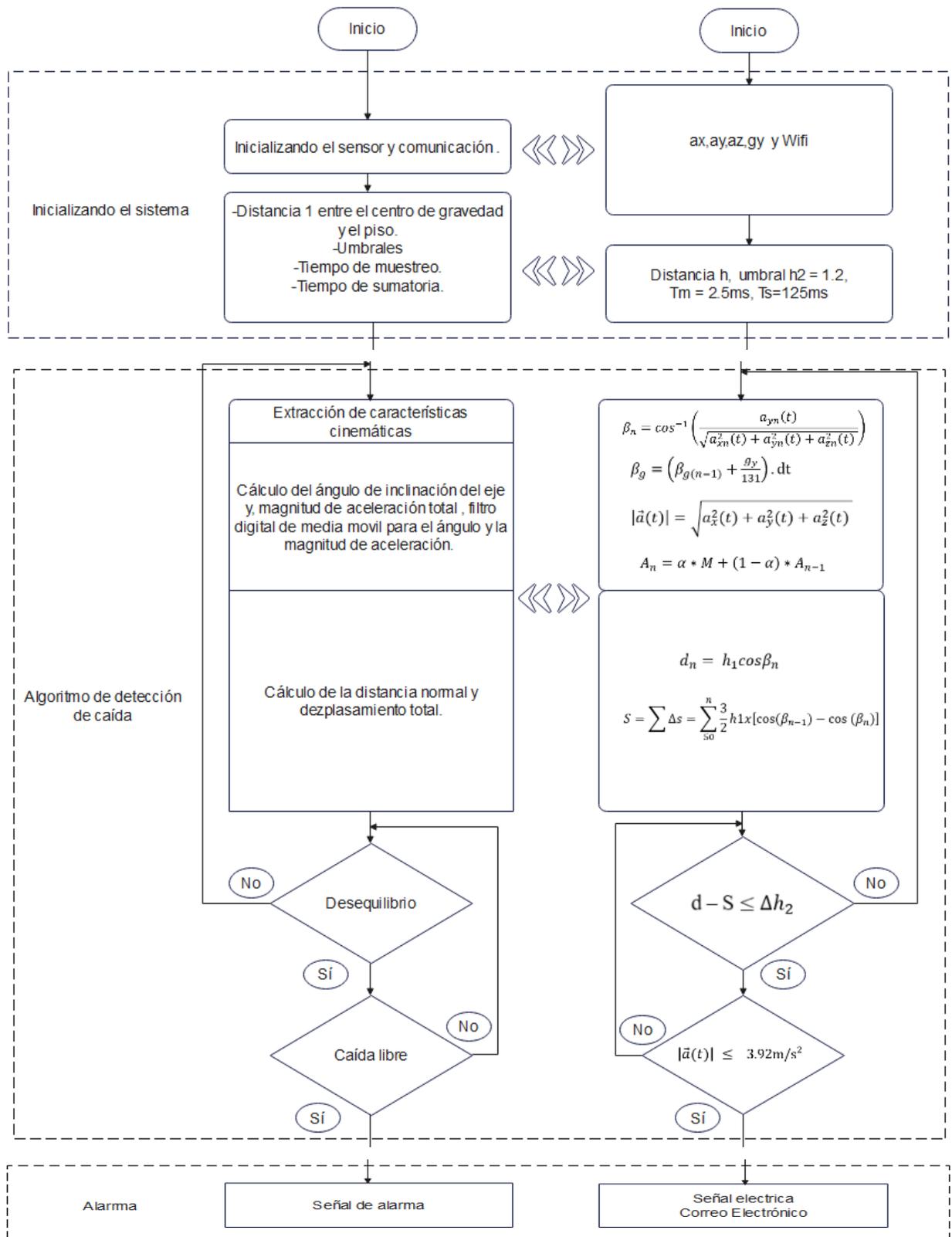


Figura 4.8 Diagrama de flujo de detección de una caída.

4.4.4 Diseño del sistema de sujeción para el sistema embebido.

Para el diseño de la estructura de sujeción se tomó las siguientes consideraciones, el suficiente espacio para el circuito mostrado en la figura 4.9 y ranuras para sus indicadores de luz, botones y una correa de nailon de una pulgada que permite la sujeción al cuerpo mediante hebillas de liberación lateral y hebillas de deslizamiento, debido a que su función es de sujetar el dispositivo sobre el centro de gravedad y el peso de este, se usó PLA.

Su estructura está constituida por una caja rectangular que cubre y sujeta mediante dos tornillos el circuito (1) y la tapa donde se sujetan los leds y botones (2) la tapa y la caja se conectan a presión con el propósito de evitar el uso de tornillos aumentado las dimensiones de la estructura.

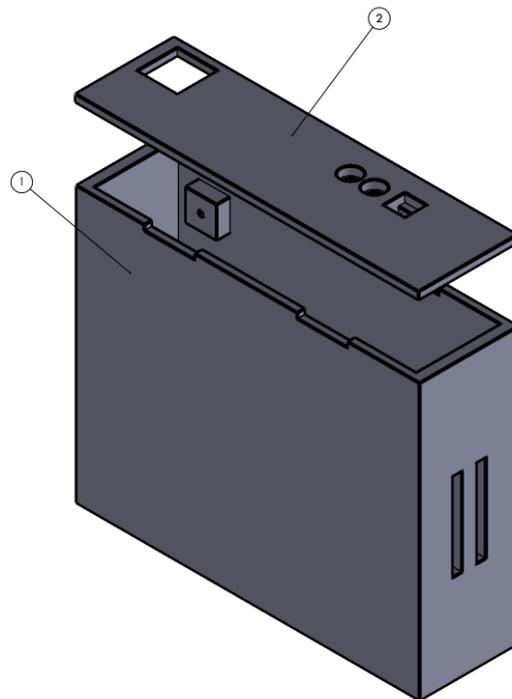


Figura 4.9 Sistema de sujeción para el sistema embebido.

4.4.5 Resultados del algoritmo.

Para evaluar el algoritmo se usó una matriz de confusión representada en la tabla 4.5 donde se tomó cuatro muestras por cada actividad en la primera columna se observa las actividades que se realizaron basadas en otros trabajos del mismo tipo y en las columnas continuas están su resultado que pueden ser cuatro casos posibles para su comprensión tomaremos las siguientes premisas:

- Dato real es la actividad que está aconteciendo en la vida real.
- Dato detectado es la actividad que el algoritmo detecto.
- Verdadero cuando la actividad es una caída.
- Falso cuando la actividad no es una caída.

Casos:

Verdadero positivo (VP) cuando el dato real es verdadero y el dato detectado es verdadero.

Verdadero negativo (VN) cuando el dato real es falso y el dato detectado es falso.

Falso positivo (FP) cuando el dato real es falso y el dato detectado es verdadero.

Falso negativo (FN) cuando el dato real es verdadero y el dato detectado es verdadero.

Actividad	VP	VN	FP	FN
Agacharse		12		
Levantar un objeto		12		
Caminar		12		
Acostarse/levantarse		11	1	
Sentarse		12		
Subir las escaleras/Bajar las escaleras		12		
Sentadilla		12		
Caída de rodilla	1			11
Caída Frontal	11			1
Caída Posterior	12			0
Caída lateral	8			3
Caída Frontal Flexionando Rodilla	10			2
Caída Posterior Flexionando Rodilla	8			4
Caída lateral izquierdo Flexionando Rodilla	8			4
Caída lateral derecha Flexionando Rodilla	9			3
Total 1	67	83	1	28
Total 2	31	83	1	4

Tabla 4.5 Matriz de confusión del algoritmo de detección de caídas sin flexionar las rodillas.

Los resultados obtenidos a partir de pruebas experimentales donde un grupo de seis personas realizaron cada actividad cuatro veces donde tres personas de la tercera edad realizaron las actividades comunes y tres personas jóvenes realizaron las actividades de caídas los resultados se muestran de la matriz de confusión se muestran en la tabla 4.6 donde se obtuvo la sensibilidad, especificidad y precisión a partir de las ecuaciones 2.10, 2.11 y 2.12.

Se obtienen dos resultados el primero se obtuvo a partir del método donde se evalúa el algoritmo con todas las actividades propuestas en la tabla 4.5, el segundo resultado se obtuvo evaluando el algoritmo con caídas sin flexionar las rodillas.

Comprobando así que debido a la flexión de las rodillas durante una caída el algoritmo redujo su sensibilidad como se observa en la tabla 4.6.

	Resultado 1	Resultado 2
Sensibilidad	70,53	88,57%
Especificidad	98,81	98,81%
Precisión	84,67	93,69%

Tabla 4.6 Resultados del algoritmo de detección de caídas sin flexionar las rodillas.

4.4.6 Tiempo de espera.

Para determinar el tiempo de espera se usó un cable de cinco metros USB debido a que el microcontrolador no puede enviar datos a 2.5 ms de forma inalámbrica y CoolTerm una aplicación que guarda los datos obtenidos del puerto serial los datos registrados se pueden observar dentro del Anexo B.1.

En la figura 4.10 se graficó los datos obtenidos durante una caída sin flexionar las rodillas, donde el eje y es la aceleración sus unidades son m/s^2 , el eje x es el tiempo sus unidades son ms. El primer valor indica que detecto una caída y su tiempo en una aceleración menor al umbral y el segundo valor indica el tiempo donde se registró la aceleración máxima ya que esto indica que el cuerpo impacto y posteriormente al estar acostado su aceleración disminuye.

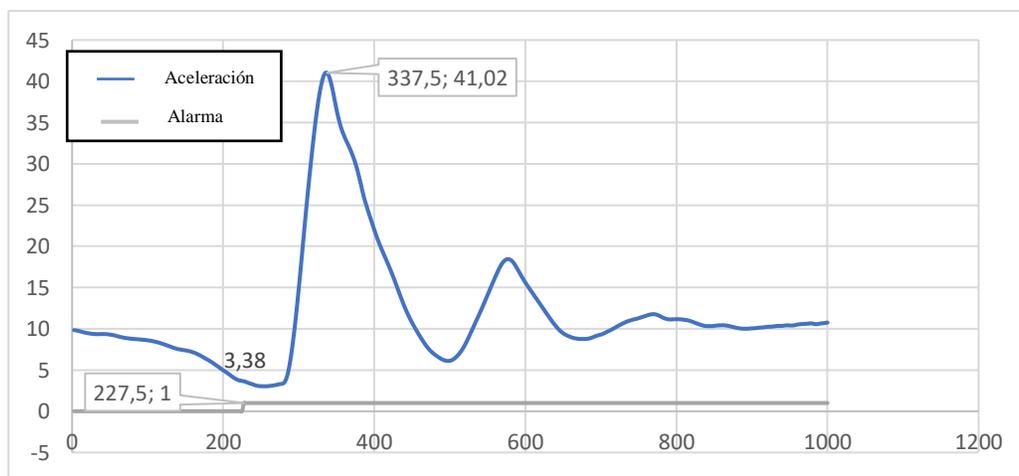


Figura 4.10 Muestra de una caída sin flexionar las rodillas.

Para obtener el tiempo de espera se resta el tiempo de alarma con el tiempo de aceleración máxima dando un promedio de 107,5 ms como muestra la tabla 4.7.

Tiempo de alarma (ms)	Tiempo de aceleración máximo (ms)	Tiempo de espera (ms)
202,5	305	102,5
227,5	337,5	110
215	325	110
PROMEDIO		107,5 ms

Tabla 4.7 Muestras registradas.

4.5 Diseño del algoritmo para detectar caídas flexionando las rodillas.

Para determinar este tipo de caídas se realiza un estudio de aceleración y desplazamiento angular en la rodilla mediante un sistema embebido donde se recopiló datos en la actividad de agacharse apoyando las rodillas para obtener los umbrales debido a que usando solo el umbral de desequilibrio con un tiempo de desplazamiento de 12,5 ms se obtuvo falsos positivos en la actividad de agacharse pero mejorando su sensibilidad hasta un 75% y con el umbral de aceleración mínimo de 3.92 m/s^2 se obtuvo falsos negativos los resultados obtenidos se muestra en la matriz de confusión en anexos.

4.5.1 Diseño del sistema embebido para la rodilla.

El circuito se observa en la figura 4.11 está compuesto por el microcontrolador ESP- 8266 D1 MINI se encarga de recibir los datos de un sensor MPU-6050 procesar, filtrar la señal del sensor y calcular la aceleración, desplazamiento angular total y la distancia normal este dispositivo está energizado con una batería de 3.7v y 6800mAh.

4.5.2 Selección de umbrales de desequilibrio y caída libre para la rodilla.

Se realizó la adquisición de datos con el mismo método realizado para el cálculo de tiempo de espera con las actividades de agacharse y caídas flexionando la rodilla.

Umbral de desequilibrio.

Para obtener el umbral de desequilibrio se realizó el análisis de los datos obtenidos del desplazamiento total en un periodo de tiempo de 12,5 ms este tiempo se obtuvo a partir de pruebas experimentales se pueden observar en anexos B.2, para la obtención del umbral se buscó una relación de la diferencia entre la distancia normal y el desplazamiento total, se usó la distancia normal para reconocer el impacto de la persona sobre el suelo ya que esta tiende a ser cero.

En la figura 4.12 se puede observar los datos obtenidos de una muestra de una actividad de caída flexionando las rodillas en el eje y indica el desplazamiento normal en unidades de cm y en el eje x se muestra el tiempo transcurrido en ms.

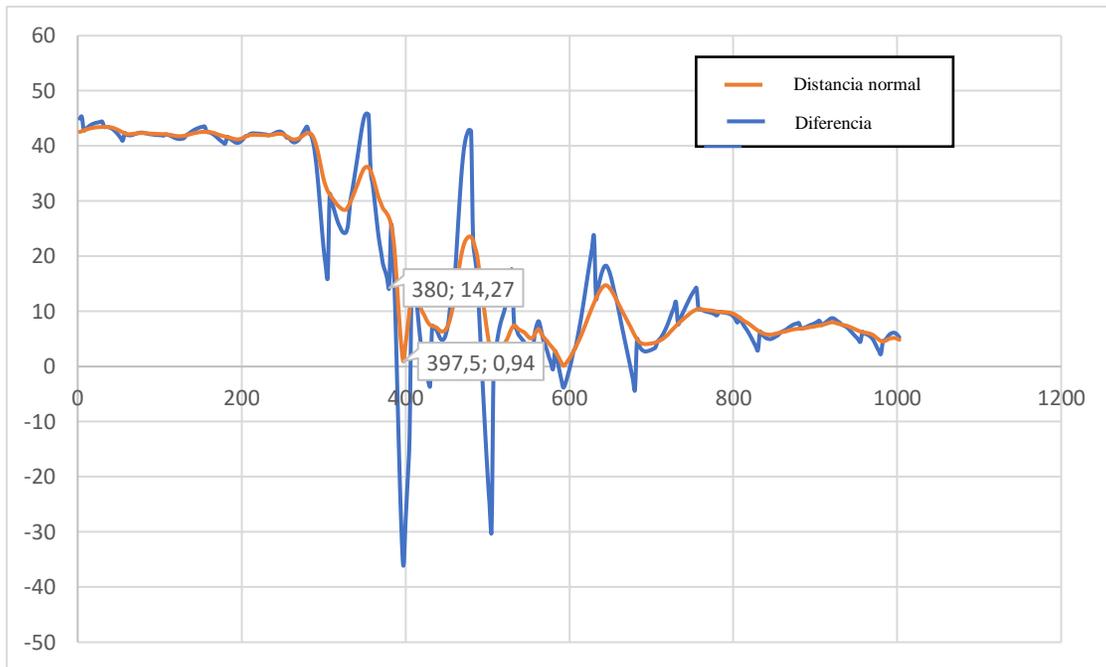


Figura 4.12 Muestra de una actividad de caída flexionando las rodillas.

La línea tomate es la distancia normal y la línea azul es la diferencia entre la distancia normal y el desplazamiento el valor de 14,27 se tomo como referencia para el umbral mediante pruebas experimentales se llegó al umbral de 3,2.

Umbral de caída libre.

En la figura 4.13 se puede observar los datos obtenidos de una muestra de una actividad de agacharse apoyando las rodillas en el eje y indica la aceleración en unidades de m/s^2 y en el eje x se muestra el tiempo transcurrido en ms.

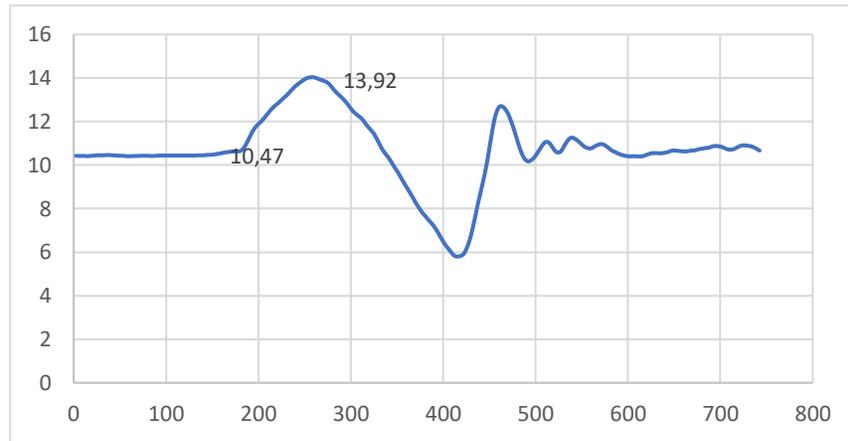


Figura 4.13 Muestra de aceleración de la actividad de agacharse apoyando las rodillas.

En la tabla 4.8 se muestra los valores obtenidos de cinco muestras se pueden observar en Anexos B.3 de su aceleración mínima antes del impacto que se reconoce por la aceleración máxima registrada, dando un promedio de aceleración mínima de 10,57 y su promedio en aceleración máxima de 13,96.

N° Muestra	Mínima	Máxima
1	10,47	13,92
2	10,50	13,98
3	10,75	14,05
4	10,46	13,45
5	10,65	14,42
Promedio	10,57	13,96

Tabla 4.8 Aceleración mínima y máxima de cinco muestras de la actividad de agacharse.

En la figura 4.14 se puede observar los datos obtenidos de una muestra de una actividad de caída flexionando las rodillas en el eje y indica la aceleración en unidades de m/s^2 y en el eje x se muestra el tiempo transcurrido en ms.

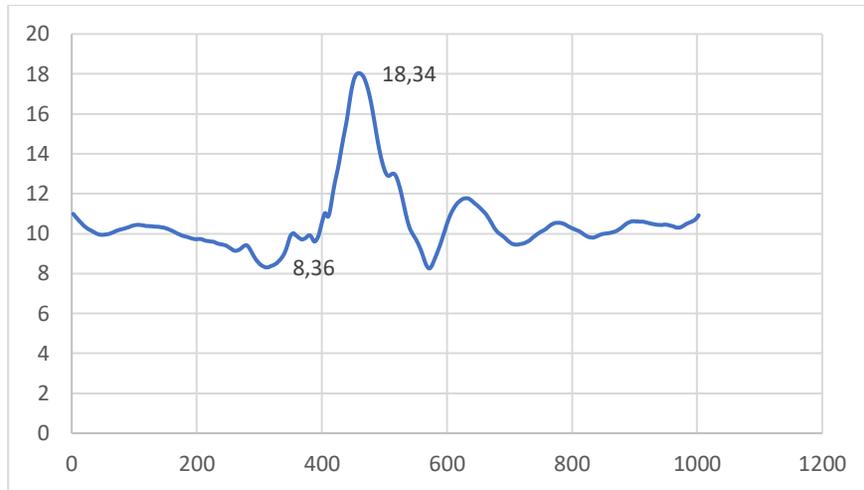


Figura 4.14 Muestra de una actividad de caída flexionando las rodillas

En la tabla 4.9 se muestra los valores obtenidos de cinco muestras sus resultados se pueden observar en Anexos B.4 de su aceleración mínima antes del impacto que se reconoce por la aceleración máxima registrada, dando un promedio de aceleración mínima de 8,2 y su promedio en aceleración máxima de 20,63.

N° Muestra	Mínima	Máxima
1	8,36	18,34
2	8,31	20,58
3	8,04	21,45
4	7,88	21,07
5	8,47	21,72
Promedio	8,21	20,63

Tabla 4.9 Aceleración mínima y máxima de cinco muestras de la actividad de caída flexionando las rodillas.

Con los umbrales seleccionado se usó un umbral de aceleración mínima de 8,5 ya que la aceleración mínima más alta registrada fue de 8,47 se usó la aceleración mínima para tener un tiempo entre la detección de caída y el impacto.

4.5.3 Algoritmo para las caídas flexionando la rodilla.

En la figura 4.15 muestra un diagrama donde se muestra los puntos donde se realizó el análisis de desequilibrio y caída libre en el punto (p1) indica la posición del circuito uno donde se implementa

el algoritmo para la detección de caídas sin flexionar la rodilla a la altura del centro de gravedad, punto dos (p2) indica la posición del circuito dos a la altura de la rodilla donde se implementa el algoritmo sin flexionar las rodillas (ver la figura 4.16) para su implementación se programó en el lenguaje C del IDE de Arduino, dentro del **Anexo C** se presenta el código de programación cargado a cada dispositivo.

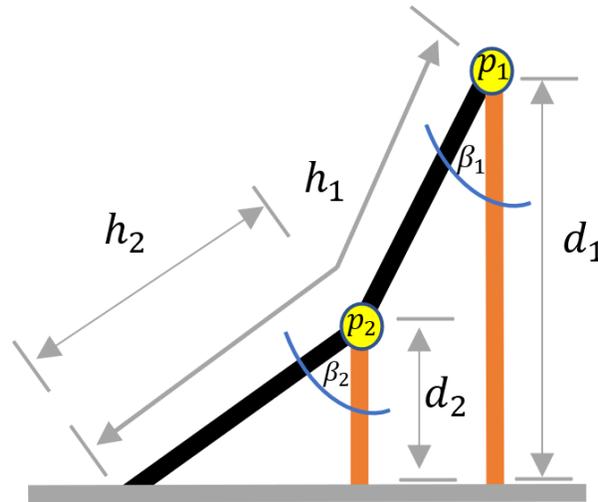


Figura 4.15 Diagrama de una caída.

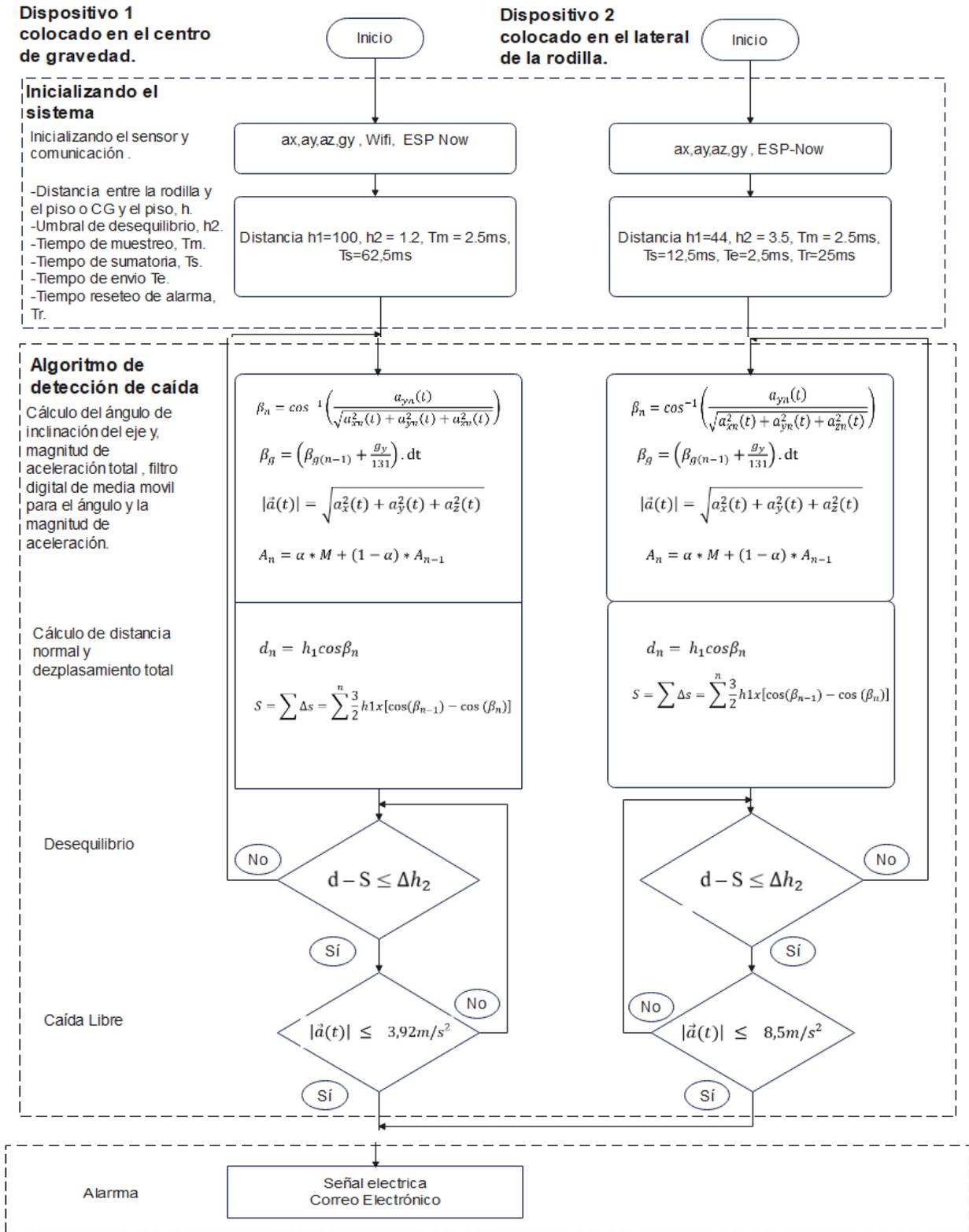


Figura 4.16 Diagrama de flujo para el dispositivo uno y dos.

4.5.4 Diseño de la red inalámbrica usando ESP-Now

Usando el protocolo de comunicación inalámbrica ESP – Now se configura, el microcontrolador ESP8266 como emisor ya que al detectar una caída envía la alarma al microcontrolador ESP-32 (ver figura 4.17) para que envíe el correo ya que la función de correo ocupa demasiado espacio en la RAM el microcontrolador ESP8266 no puede instalarse esta función, mediante el programa en el anexo A.3 es utilizado para conocer el MAC del receptor.

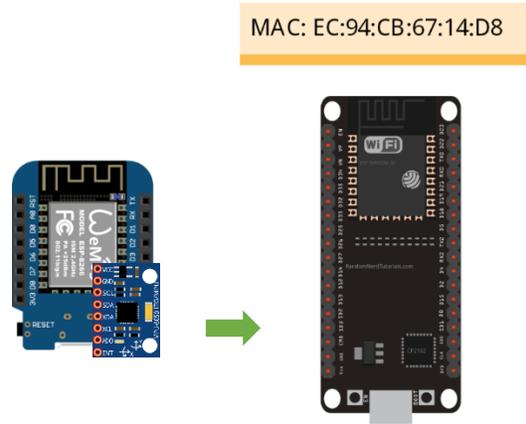


Figura 4.17 Configuración de red.

4.5.5 Diseño del sistema de sujeción de los sensores.

Como se observa en la figura 4.18 su estructura está constituido por una caja que sujeta mediante un tornillo el circuito con compartimiento para la batería (1) y ranuras para una correa de nylon de una pulgada, las tapas (2) (3) sirven para cubrir al mismo.

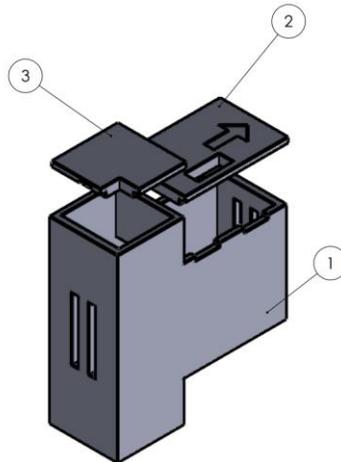


Figura 4.18 Estructura de sujeción del sistema embebido y el cuerpo.

4.5.6 Adquisición de datos para obtener el tiempo de espera.

Con los dispositivos conectados al cuerpo como se muestra en la figura 4.19 se tomó muestras de la aceleración y la activación de la señal de alarma de una caída flexionando las rodillas y sin flexionar las rodillas con el propósito de determinar el tiempo de espera.

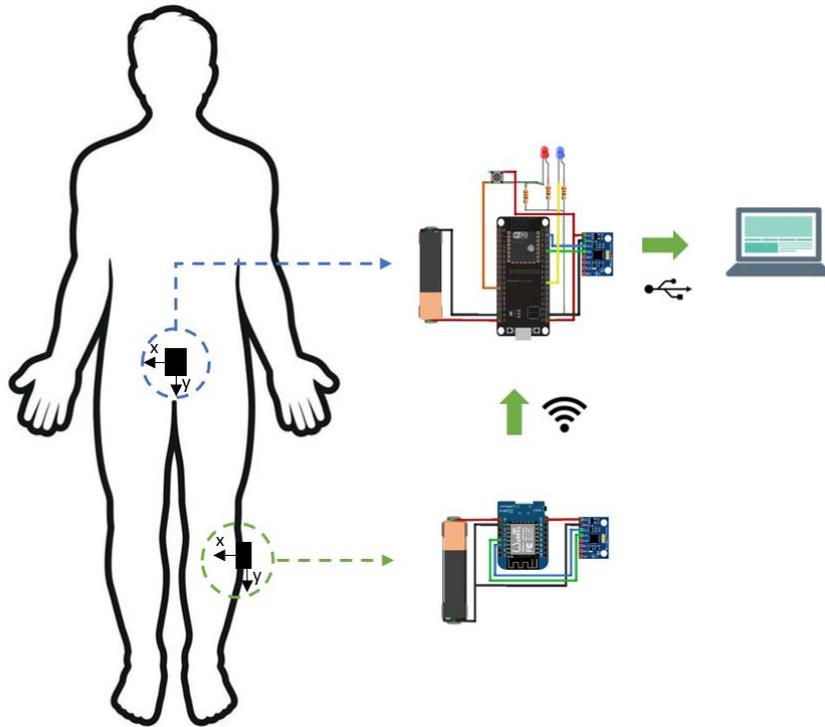


Figura 4.19 Ubicación y envío de datos de los dispositivos para la adquisición de datos.

Se obtuvieron dos tiempos de espera el primero que es detectado por el dispositivo colocado en el centro de gravedad que se encarga de detectar las caídas sin flexionar las rodillas que se puede diferenciar ya que se activa la alarma con una aceleración menor al umbral de 3.92 m/s^2 como muestra la figura 4.20.

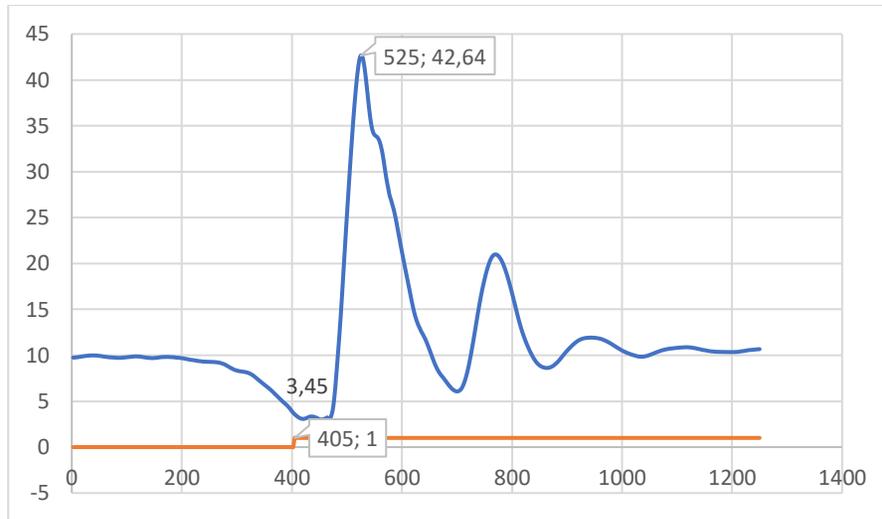


Figura 4.20 Muestra de caída sin flexionar la rodilla.

Analizando cinco muestras como se observa en la tabla 4.10 el tiempo de espera es en promedio 116,4 ms y además sus demás resultados se pueden observar en Anexos B.5.

Tiempo de alarma (ms)	Tiempo de aceleración máximo (ms)	Tiempo de espera (ms)
405,5	525	119,5
362,5	485	122,5
360	470	110
555	665	110
490	610	120
PROMEDIO		116,4 ms

Tabla 4.10 Tabla de resultados de las muestras de caídas sin flexionar las rodillas.

El tiempo de espera para el dispositivo colocado en la rodilla que se encarga de detectar las caídas flexionando la rodilla se puede diferenciar mediante la aceleración que se registró en la activación de la alarma que es mayor al del umbral $3,92\text{m/s}^2$ como se observa en la figura 4.21.

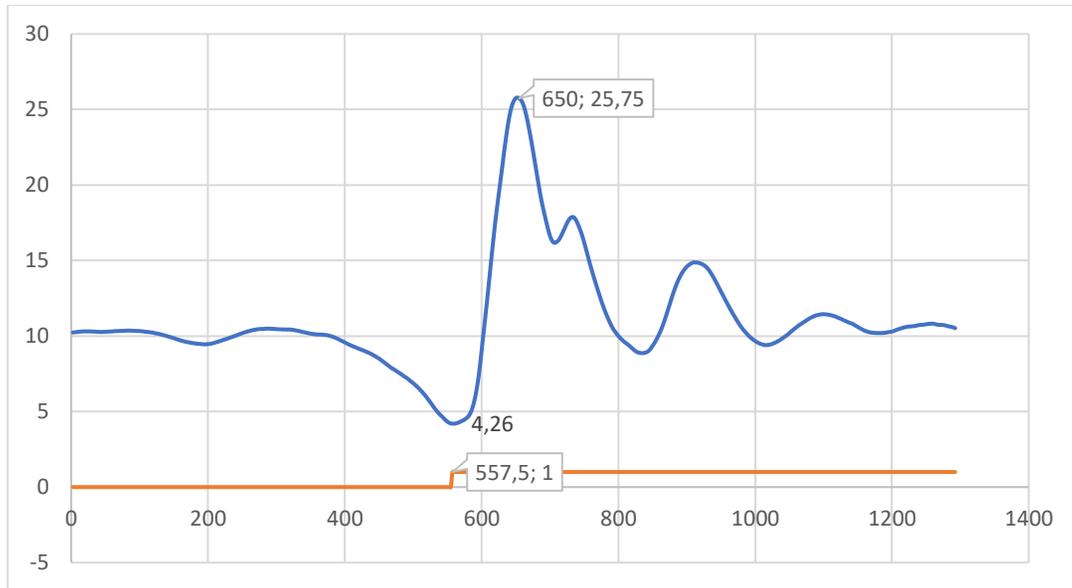


Figura 4.21 Muestra de caída flexionando las rodillas.

Analizando cinco muestras como se observa en la tabla 4.11 y sus demás muestras se encuentran en Anexos B.6 el tiempo de espera es en promedio 99,8 ms.

Tiempo de alarma (ms)	Tiempo de aceleración máximo (ms)	Tiempo de espera (ms)
557,5	650	92,5
385	472,5	87,5
387	480	92,5
385	521,5	136,5
295	385	90
PROMEDIO		99,8 ms

Tabla 4.11 Tabla de resultados de las muestras de caídas flexionando las rodillas.

4.5.7 Matriz de confusión

Aplicando las actividades para evaluar el algoritmo en un grupo de diez personas, cinco personas de la tercera edad realizaron las actividades comunes y cinco personas jóvenes las actividades de caídas donde cada una repite cuatro veces cada actividad los resultados se observan en la matriz de confusión que se observa en la tabla 4.12 da como resultado 97,50% sensibilidad, 87,12% de especificidad y 92,31% de precisión.

Actividad	VP	VN	FP	FN
Agacharse			8	4
Levantar un objeto			20	
Caminar			20	
Acostarse/levantarse			9	11
Sentarse			20	
Subir las escaleras/bajar las escaleras			18	2
Sentadilla			20	
Caída de rodilla		18		2
Caída Frontal		19		1
Caída Posterior		20		
Caídas laterales		20		
Caída Frontal Flexionando Rodilla		20		
Caída Posterior Flexionando Rodilla		20		
Caída lateral izquierdo Flexionando Rodilla		20		
Caída lateral derecha Flexionando Rodilla		19		1
Total		156	112	20

Tabla 4.12 Matriz de confusión

Las actividades con un índice notable de falsos positivos son agacharse, acostarse, levantarse y subir las escaleras como se puede observar en la figura 4.22.



Figura 4.22 Actividades con un mayor índice de falsos positivos a) Agacharse con apoyo de rodilla, b) Subir/bajar escaleras, c) Acostarse y levantarse.

Estas actividades difieren en sus resultados dependiendo de las patologías de la persona en caso de personas que no son consideradas como personas de la tercera edad, pero tiene problemas de desgaste de cartílago en rodilla existe un mayor índice de falsos positivos debido a que la rigidez de su rodilla provoca que la persona realice movimientos más rápidos de rodilla para llegar a la posición donde se sienten más seguros.

Para personas de la tercera edad este índice de falsos positivos disminuye a edades más avanzadas a partir de los 79 años debido a la condición física, problemas en sus articulaciones provocan movimientos más lentos, no pueden realizar ciertos movimientos como agacharse doblando las rodillas y la sentadilla o también la manera de como realizan el movimiento como muestra la figura 4.23 apoyan solo una rodilla y bajan las escaleras con un pie a la vez.



Figura 4.23 Actividad de agacharse.

4.5.8 Ajustes y correcciones.

Se realizan las correcciones pertinentes enfocándose en la actividad de acostarse y levantarse ya que las anteriores pierden importancia debido a que las personas de edad más avanzada no realizan estos movimientos o lo hacen de manera más lenta.

Existe un índice alto de falsos positivos en la actividad de acostarse y levantarse debido que al acostarse se suspenden en el aire las piernas como se observa en la figura 4.24 en el punto B la zona cercana de la rodilla d tiende a cero porque el ángulo Θ tiende a 90° por ende la sumatoria del desplazamiento tiene un mayor valor con cualquier movimiento realizado lo que permite que se cumpla con la condición de desequilibrio y la aceleración tienda a cero debido a la aceleración opuesta a la gravedad representada en el grafico con la letra “a” que se produce por la fuerza que realiza la persona al levantar las piernas lo que detecta el algoritmo como caída libre para rectificar estos falsos positivos se puede observar que en el punto A al existir el apoyo sobre la cama en todo

momento de esta actividad por lo general la única aceleración ejercida sobre este punto es la aceleración de la gravedad por ende se implementa otro umbral de aceleración de 9 m/s^2 que determina si la persona se encuentra en equilibrio.

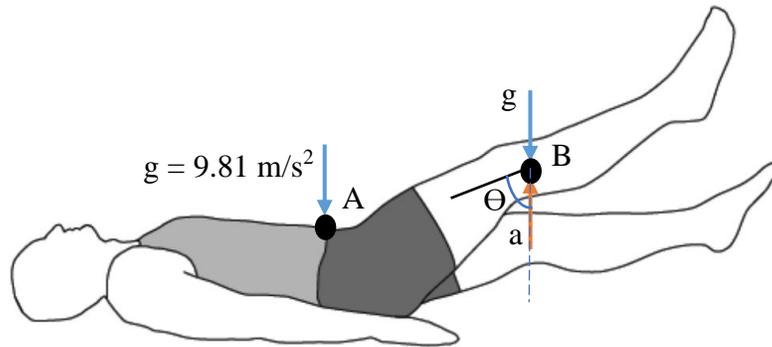


Figura 4.24 Cinemática de la actividad de acostarse.

El algoritmo de detección de caídas con las correcciones realizadas se puede observar en la figura 4.25 representado como un diagrama de flujo.

Dispositivo 1
colocado cerca del
centro de gravedad.

Dispositivo 2
colocado en el lateral
de la rodilla.

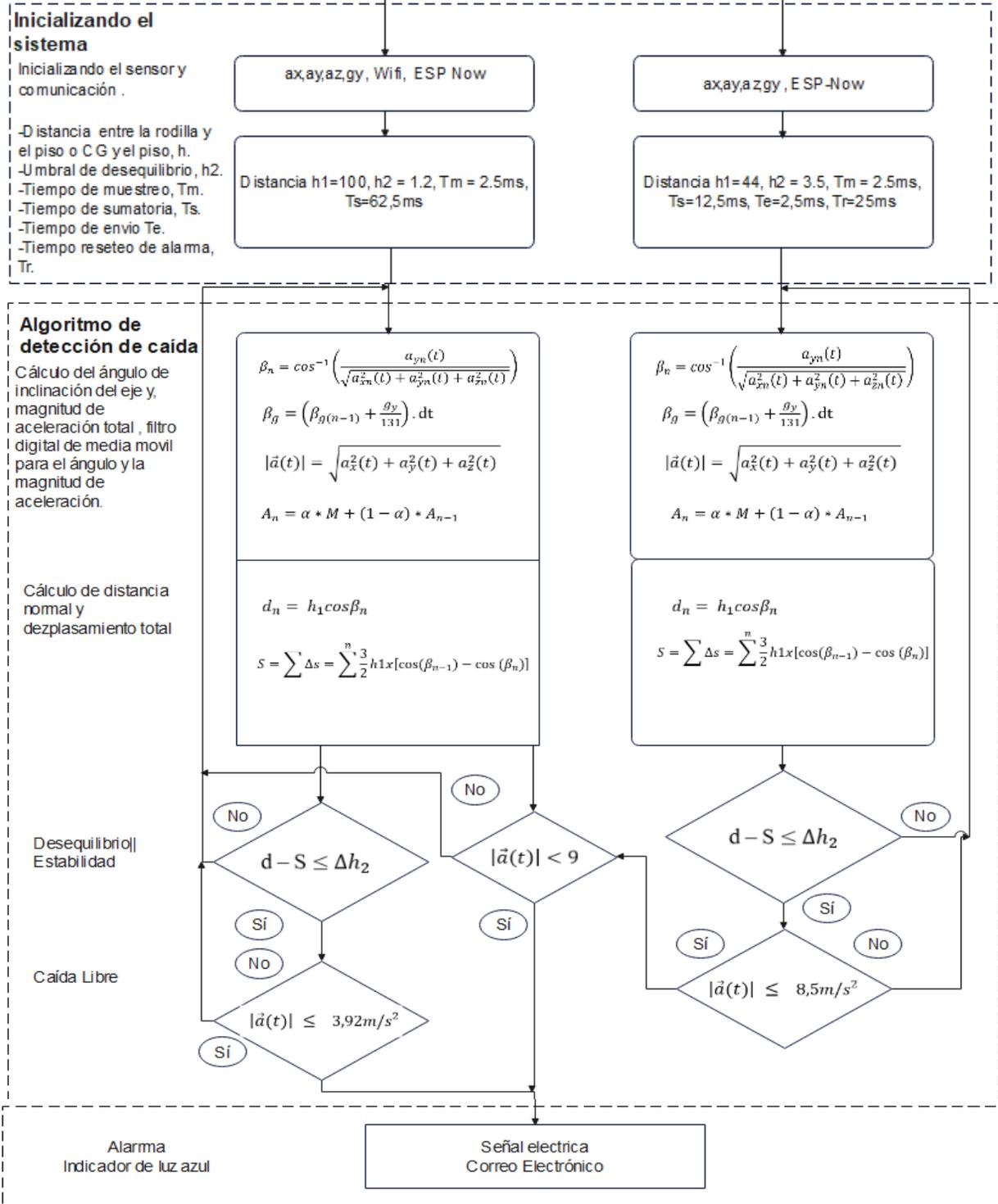


Figura 4.25 Algoritmo de detección de caídas.

A partir de pruebas las pruebas experimentales se obtuvo los resultados mostrados en la tabla 4.13.

Actividad	VP	VN	FP	FN
Agacharse			20	
Levantar un objeto			20	
Caminar			20	
Acostarse/levantarse			20	
Sentarse			20	
Subir las escaleras/Bajar las escaleras			17	3
Sentadilla			20	
Caída de rodilla		18		2
Caída Frontal		19		1
Caída Posterior		19		1
Caída Lateral		20		
Caída Frontal Flexionando Rodilla		19		1
Caída Posterior Flexionando Rodilla		20		
Caída Lateral Flexionando Rodilla		20		
Total 1		135	137	3
				5

Tabla 4.13 Matriz de confusión.

El algoritmo obtuvo una sensibilidad del 96,43%, especificidad del 97,86% y una precisión del 97,14%.

4.5.9 Implementación del protocolo simple de transferencia de correo.

Para este sistema para él envió de correo se utiliza una cuenta Gmail y la librería para Arduino (esp mail client), para su funcionamiento se activa la clave de aplicación, se configura las credenciales de la red de internet usuario y contraseña, del remitente correo electrónico y contraseña de aplicación, del remitente correo electrónico por ultimo para el protocolo smtp se define el host y el puerto (ver figura 4.26).

```
#define WIFI_SSID "Nombre de red"
#define WIFI_PASSWORD "Constraseña"

#define SMTP_HOST "smtp.gmail.com"
#define SMTP_PORT 465

#define AUTHOR_EMAIL "Correo del remitente"
#define AUTHOR_PASSWORD "contraseña de aplicacion"

#define RECIPIENT_EMAIL "Correo del receptor"
```

Figura 4.27 Programación en el IDE de Arduino de la configuración del protocolo SMTP.

Se configura la función que se encarga de enviar el mensaje como se muestra en la figura 4.28 donde se ingresa las credenciales, nombre del remitente, asunto y mensaje.

```
void correo (){
  SMTPSession smtp;
  smtp.debug(1);
  ESP_Mail_Session session;
  session.server.host_name = SMTP_HOST;
  session.server.port = SMTP_PORT;
  session.login.email = AUTHOR_EMAIL;
  session.login.password = AUTHOR_PASSWORD;
  session.login.user_domain = "";
  SMTP_Message message;
  message.sender.name = "Nombre del remitente";
  message.sender.email = AUTHOR_EMAIL;
  message.subject = "Asunto";
  message.addRecipient("Familiar", RECIPIENT_EMAIL);
  String htmlMsg = "<div style=\"color:#2f4468;\"><h1>Mensaje a enviar!</h1><p>- .....</p></div>";
  message.html.content = htmlMsg.c_str();
  message.html.content = htmlMsg.c_str();
  message.text.charset = "us-ascii";
  message.html.transfer_encoding = Content_Transfer_Encoding::enc_7bit;
  if (!smtp.connect(&session)){
    return;}
  if (!MailClient.sendMail(&smtp, &message)){
    Serial.println("Error sending Email, " + smtp.errorReason());
    delay(10000);
  }
}
```

Figura 4.28 Programación en el IDE de Arduino de la función que se encarga de enviar el correo.

El formato del mensaje recibido por el receptor se muestra en la figura 4.29.



Figura 4.29 Visualización del correo recibido.

CONCLUSIONES

El algoritmo presentado es capaz de detectar caídas con o sin flexionar las rodillas, logrando una precisión del 97,14% con un tiempo entre la activación de la alarma y el impacto de la caída de 99,8 ms para caídas flexionando las rodillas y de 116,4 ms sin flexionarlas permitiendo la implementación de sistemas de amortiguación de caídas que funcionen en esos periodos de tiempo, además de que puede ser implementado sobre dispositivos de bajo costo, en caso de detectar una caída envía un correo electrónico a su cuidador para dar la atención medica de forma inmediata y así evitar las complicaciones de no ser atendido a tiempo.

El algoritmo se validó mediante una matriz de confusión que tomo como actividades evaluativas las más comunes que puede realizar una persona y caídas flexionando y sin flexionar las rodillas, este método evaluativo además de calificar al sistema permite identificar las actividades que tienden a ocasionar errores en su precisión, así enfocarse en su estudio y ajustar los umbrales en menor tiempo.

Para el diseño de este sistema se tomó en cuenta las fases de una caída y su relación que tienen estas con la cinemática de una caída y el comportamiento de la aceleración durante este accidente tomando como ubicación de estudio dos partes del cuerpo su centro de gravedad y la rodilla esta última debido a la acción involuntaria de las personas de flexionar las rodillas, dando como resultado que los cambios bruscos del desplazamiento en un periodo de tiempo y la tendencia a cero de la distancia normal determinan el desequilibrio siendo esta la primera fase de la caída, la aceleración mínima registrada durante una caída se relaciona con la segunda fase la caída libre, permitiendo que el sistema tenga un tiempo entre la activación de la alarma y el impacto, aunque si se desea una mayor precisión del sistema se puede usar la aceleración máxima ya que los valores máximos de una caída son muy característicos a diferencia de las otras actividades y esta se relaciona con la tercera fase de la caída el impacto de la persona al suelo pero este valor limitaría las funciones que puede cumplir un algoritmo portable de no poder implementar un sistema de amortiguación de impacto ya que no existiría un tiempo de espera. Debido a que las extremidades tienen como característica tener un alto rango de amplitud de movimientos el dispositivo implementado en la rodilla tiende a presentar demasiados falsos positivos, para mejorar su especificidad se toma en cuenta si el dispositivo ubicado en el centro de gravedad solo esta sometido a la fuerza de gravedad esto determina que la zona del tronco está en equilibrio, de no ser el caso toma como verdadero la detección de caída detectado por el dispositivo en la rodilla.

RECOMENDACIONES

Para futuras investigaciones para incorporar el sistema de amortiguación de impacto de la caída es necesario mejorar el tiempo de respuesta de este, permitirá la implementación de más métodos de sistemas de amortiguación y no limitarse por el tiempo de respuesta que tiene actualmente.

Para su implementación sobre personas es necesario diseñar el hardware teniendo en cuenta reducir sus dimensiones lo máximo posible para no incomodar al usuario y sea cómodo su uso, también incorporar métodos de ahorro de energía además de aumentar la autonomía del dispositivo evitando la preocupación de cargar las baterías con frecuencia reducirá sus dimensiones por el uso de baterías de menor corriente y por ende de menor tamaño.

Incorporar un método de adquisición de datos sin el uso de cables permitirá mayor facilidad de estudiar los movimientos y además de en caso de ya ser implementado sobre una persona registrar datos reales de una caída.

Realizar más estudios para incorporar más actividades que el sistema puede detectar permitirá que el algoritmo no funcione solo para personas de la tercera edad que realizan movimientos más lentos también serviría para personas que tenga un ritmo de vida más activo pero que sufran otras patologías que producen caídas.

REFERENCIAS

- [1] MIES, “Informe mensual de gestión del servicio de atención y cuidado para personas adultas mayores y del estado situacional de su población,” May 2020.
- [2] D. Jannina and E. M. Medina, “Evaluación del riesgo de caídas en adultos mayores que reciben intervención con un programa multifactorial de prevención comparado con adultos mayores que reciben indicaciones en consulta médica del centro de salud número 9 comité del pueblo en los meses de septiembre a noviembre de 2016,” 2016.
- [3] G. Oumaima, A. A. Hamd, T. Youness, O. H. T. Rachid, and B. Omar, “Vision-based fall detection and prevention for the elderly people: A review ongoing research,” *5th International Conference on Intelligent Computing in Data Sciences, ICDS 2021*, 2021, doi: 10.1109/ICDS53782.2021.9626736.
- [4] A. Carballo-Rodríguez, J. Gómez-Salgado, I. Casado-Verdejo, B. Ordás, and D. Fernández, “Estudio de prevalencia y perfil de caídas en ancianos institucionalizados,” *Gerokomos*, vol. 29, no. 3, pp. 110–116, 2018.
- [5] R. González Vega, R. Hernández Bretones, B. Jiménez del Olmo, R. González Vega, R. Hernández Bretones, and B. Jiménez del Olmo, “Desarrollo de un sistema de detección de caídas basado en acelerómetros,” 2016.
- [6] S. J. Barber, K. Hamel, C. Ketcham, K. Lui, and N. Taylor-Ketcham, “The effects of stereotype threat on older adults’ walking performance as a function of task difficulty and resource evaluations.,” *Psychol Aging*, vol. 35, no. 2, pp. 250–266, Mar. 2020, doi: 10.1037/pag0000440.
- [7] J. Li, “Wearable and controllable protective system design for elderly falling,” in *Proceedings - 2020 6th International Conference on Mechanical Engineering and Automation Science, ICMEAS 2020*, Oct. 2020, pp. 187–194. doi: 10.1109/ICMEAS51739.2020.00042.
- [8] Y. Ning, Y. Diao, G. Wang, N. Lou, G. Li, and G. Zhao, “Automatic Fall Protection for Hips Based on Micromechanical Double Gas Cylinder Rapid Puncture and Bionic Capsule Inflation,” in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2021, pp. 6783–6786. doi: 10.1109/EMBC46164.2021.9630780.
- [9] S. Bhatlawande, D. Khapre, M. Kinge, and T. Khairnar, “Vision Based Assistive System for Fall Detection,” in *2022 2nd International Conference on Intelligent Technologies, CONIT 2022*, 2022. doi: 10.1109/CONIT55038.2022.9847697.
- [10] T. Kalinga, C. Sirithunge, A. G. Buddhika, P. Jayasekara, and I. Perera, “A Fall Detection and Emergency Notification System for Elderly,” in *2020 6th International Conference on Control, Automation and Robotics, ICCAR 2020*, Apr. 2020, pp. 706–712. doi: 10.1109/ICCAR49639.2020.9108003.
- [11] L. Ren and Y. Peng, “Research of fall detection and fall prevention technologies: A systematic review,” *IEEE Access*, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 77702–77722, 2019. doi: 10.1109/ACCESS.2019.2922708.
- [12] Badillo Emmanuel, “Manual de la asignatura de Biomecánica aplicada al Deporte,” Universidad Autónoma del estado de Morelos, Morelos, 2017.
- [13] J. Hamill, K. Knuzen, and T. Derrick, *Biomecánica Bases del movimiento humano*, Wolter Kluvers., vol. 4. 2017.

- [14] S. Usmani, A. Saboor, M. Haris, M. A. Khan, and H. Park, “Latest research trends in fall detection and prevention using machine learning: A systematic review,” *Sensors*, vol. 21, no. 15. MDPI AG, Aug. 01, 2021. doi: 10.3390/s21155134.
- [15] F. A. S. Ferreira De Sousa, C. Escriba, E. G. Avina Bravo, V. Brossa, J. Y. Fourniols, and C. Rossi, “Wearable Pre-Impact Fall Detection System Based on 3D Accelerometer and Subject’s Height,” *IEEE Sens J*, vol. 22, no. 2, pp. 1738–1745, 2022, doi: 10.1109/JSEN.2021.3131037.
- [16] A. Iazzi, M. Rziza, and R. O. Haj Thami, “Fall detection system-based posture-recognition for indoor environments,” *J Imaging*, vol. 7, no. 3, 2021, doi: 10.3390/jimaging7030042.
- [17] F. Merrouche and N. Baha, “Fall detection based on shape deformation,” *Multimed Tools Appl*, vol. 79, no. 41–42, pp. 30489–30508, 2020, doi: 10.1007/s11042-019-08428-w.
- [18] T. Sivaranjani, L. DhiviyaLakshmi, R. Yogaaravinth, J. Srivishnu, M. M. S. Karthick, and A. Praveenkumar, “Fall assessment and its injury prevention using a wearable airbag technology,” in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, 2017, pp. 2539–2541. doi: 10.1109/ICPCSI.2017.8392175.
- [19] Z. Zhong *et al.*, “A Real-time Pre-impact Fall Detection and Protection System,” in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2018, pp. 1039–1044. doi: 10.1109/AIM.2018.8452687.
- [20] M. J. Flores and R. Flores, “La Enseñanza del Diagrama de Caja y Bigotes para Mejorar su Interpretación The Teaching of Box and Whiskers Plot to Improve their Interpretation,” 2018.
- [21] C. Enrique, “ALGORITMO DE DETECCIÓN DE CAÍDAS EN UN SISTEMA EMBEBIDO,” CIATEQ, Guadalajara, 2017.
- [22] T. B. Rodrigues *et al.*, “Fall detection system by machine learning framework for public health,” in *Procedia Computer Science*, 2018, vol. 141, pp. 358–365. doi: 10.1016/j.procs.2018.10.189.
- [23] J. P. Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, “Edge-AI in LoRa-based Health Monitoring: Fall Detection System with Fog Computing and LSTM Recurrent Neural Networks,” in *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, 2019, pp. 601–604. doi: 10.1109/TSP.2019.8768883.
- [24] G. Mastorakis, T. Ellis, and D. Makris, “Fall detection without people: A simulation approach tackling video data scarcity,” *Expert Syst Appl*, vol. 112, pp. 125–137, Dec. 2018, doi: 10.1016/j.eswa.2018.06.019.
- [25] M. V. Valencia and A. A. Londoño, *Introducción a la adquisición y acondicionamiento de señales*. Instituto Tecnológico Metropolitano–ITM, 2022.
- [26] H. González Allende and others, “Sistema de posicionamiento con fusión de sensores para entornos XR,” 2022.
- [27] J. F. Paz Villanueva, “Evaluación del comportamiento de voltajes y corrientes armónicas para la mejora del control en un alimentador de media tensión-casa de concesionaria de un sistema de distribución.” 2019.
- [28] D. Almaguer-Pérez and A. Hernández-Yeja, “Buenas prácticas para el despliegue seguro del servicio de correo electrónico,” *Revista cient\`ifíca*, no. 41, pp. 199–212, 2021.
- [29] N. Esteban Nieto, “Tipos de investigación,” 2018.

- [30] M. Canales, W. Paucar, and N. Juipa, “Método de investigación para ingenierías basado en la metodología de la investigación científica,” *RevIA*, vol. 7, no. 4, pp. 5–9, 2020.
- [31] M. Shen, K.-L. Tsui, M. Nussbaum, S. Kim, and F. Lure, “An Indoor Fall Monitoring System: Robust, Multistatic Radar Sensing and Explainable, Feature-Resonated Deep Neural Network,” *IEEE J Biomed Health Inform*, vol. PP, pp. 1–12, Jan. 2023, doi: 10.1109/JBHI.2023.3237077.
- [32] A. A. Rafiq, W. N. Rohman, and S. D. Riyanto, “Development of a simple and low-cost smartphone gimbal with MPU-6050 sensor,” *Journal of Robotics and Control (JRC)*, vol. 1, no. 4, pp. 136–140, Jul. 2020, doi: 10.18196/jrc.1428.
- [33] Aucatoma J, “DISEÑO E IMPLEMENTACIÓN DE UN ROBOT VELOCISTA CONTROLADO MEDIANTE WIFI UTILIZANDO UNA TARJETA DE DESARROLLO ESP32-WROOM,” Quito, Feb. 2022.
- [34] ECORFAN, “Revista de Ingeniería Eléctrica,” vol. I, Jul. 2017.
- [35] A. D. López Pozo, “Interfaz gráfica para medición de ángulos de extremidades inferiores del cuerpo humano por medio de sensores inerciales,” 2021.

ANEXOS

A. Prueba de escritorio.

Muestra	Betan-1	ang_tf	d	3/2h	COSB-		FACTOR	S	h	H2	d-S	CONDICIONAL
					l	COSB						
1	160,08	158,61	83,75	135,00	0,94	0,93	0,01	1,23	90,00	1,20	82,53	EQUILIBRIO
2	158,61	157,32	82,99	135,00	0,93	0,92	0,01	1,14	90,00	1,20	81,85	EQUILIBRIO
3	157,32	156,25	82,33	135,00	0,92	0,91	0,01	1,00	90,00	1,20	81,33	EQUILIBRIO
4	156,25	155,27	81,69	135,00	0,91	0,91	0,01	0,95	90,00	1,20	80,74	EQUILIBRIO
5	155,27	154,15	80,94	135,00	0,91	0,90	0,01	1,13	90,00	1,20	79,81	EQUILIBRIO
6	154,15	153,06	80,18	135,00	0,90	0,89	0,01	1,14	90,00	1,20	79,03	EQUILIBRIO
7	153,06	151,17	78,79	135,00	0,89	0,88	0,02	2,09	90,00	1,20	76,70	EQUILIBRIO
8	151,17	150,24	78,07	135,00	0,88	0,87	0,01	1,07	90,00	1,20	77,00	EQUILIBRIO
9	150,24	149,40	77,41	135,00	0,87	0,86	0,01	1,00	90,00	1,20	76,41	EQUILIBRIO
10	149,40	148,59	76,75	135,00	0,86	0,85	0,01	0,98	90,00	1,20	75,76	EQUILIBRIO
11	148,59	147,10	75,50	135,00	0,85	0,84	0,01	1,87	90,00	1,20	73,63	EQUILIBRIO
12	147,10	146,36	74,86	135,00	0,84	0,83	0,01	0,96	90,00	1,20	73,91	EQUILIBRIO
13	146,36	145,68	74,27	135,00	0,83	0,83	0,01	0,90	90,00	1,20	73,37	EQUILIBRIO
14	145,68	145,04	73,69	135,00	0,83	0,82	0,01	0,86	90,00	1,20	72,84	EQUILIBRIO
15	145,04	144,44	73,15	135,00	0,82	0,81	0,01	0,82	90,00	1,20	72,33	EQUILIBRIO
16	144,44	143,87	72,62	135,00	0,81	0,81	0,01	0,79	90,00	1,20	71,84	EQUILIBRIO
17	143,87	143,33	72,12	135,00	0,81	0,80	0,01	0,76	90,00	1,20	71,36	EQUILIBRIO
18	143,33	142,76	71,58	135,00	0,80	0,80	0,01	0,81	90,00	1,20	70,77	EQUILIBRIO
19	142,76	142,26	71,10	135,00	0,80	0,79	0,01	0,72	90,00	1,20	70,38	EQUILIBRIO
20	142,26	141,81	70,67	135,00	0,79	0,79	0,00	0,65	90,00	1,20	70,01	EQUILIBRIO

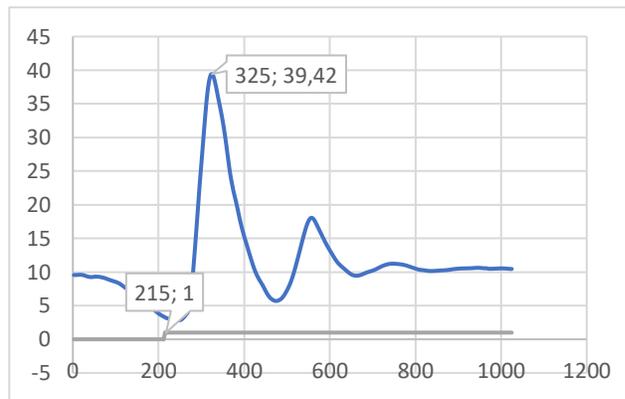
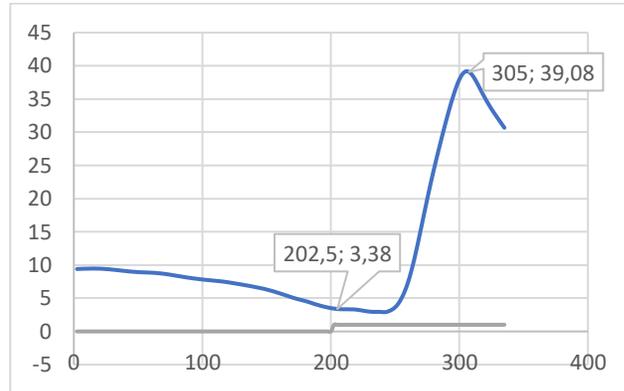
COM8

Enviar

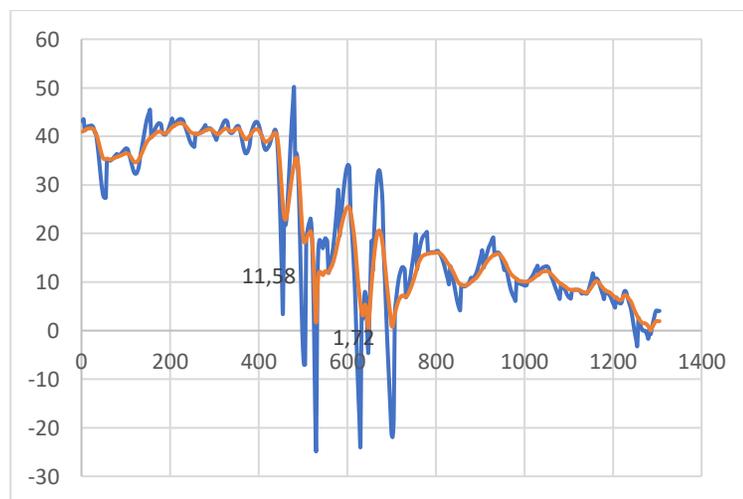
161.58	160.08	84.57	135.00	0.95	0.94	0.01	1.17	90.00	1.20	83.40
160.08	158.61	83.76	135.00	0.94	0.93	0.01	1.22	90.00	1.20	82.54
158.61	157.32	82.99	135.00	0.93	0.92	0.01	1.14	90.00	1.20	81.85
157.32	156.25	82.32	135.00	0.92	0.91	0.01	1.00	90.00	1.20	81.32
156.25	155.27	81.69	135.00	0.91	0.91	0.01	0.95	90.00	1.20	80.74
155.27	154.15	80.94	135.00	0.91	0.90	0.01	1.12	90.00	1.20	79.82
154.15	153.06	80.18	135.00	0.90	0.89	0.01	1.14	90.00	1.20	79.04
153.06	152.07	79.46	135.00	0.89	0.88	0.01	1.08	90.00	1.20	78.38
152.07	151.17	78.79	135.00	0.88	0.88	0.01	1.01	90.00	1.20	77.78
151.17	150.24	78.07	135.00	0.88	0.87	0.01	1.07	90.00	1.20	77.01
150.24	149.40	77.40	135.00	0.87	0.86	0.01	1.01	90.00	1.20	76.39
149.40	148.59	76.75	135.00	0.86	0.85	0.01	0.99	90.00	1.20	75.76
148.59	147.86	76.14	135.00	0.85	0.85	0.01	0.91	90.00	1.20	75.23
147.86	147.10	75.50	135.00	0.85	0.84	0.01	0.96	90.00	1.20	74.54
147.10	146.36	74.86	135.00	0.84	0.83	0.01	0.96	90.00	1.20	73.90
146.36	145.68	74.26	135.00	0.83	0.83	0.01	0.90	90.00	1.20	73.36
145.68	145.04	73.70	135.00	0.83	0.82	0.01	0.85	90.00	1.20	72.85
145.04	144.44	73.15	135.00	0.82	0.81	0.01	0.82	90.00	1.20	72.32
144.44	143.87	72.62	135.00	0.81	0.81	0.01	0.79	90.00	1.20	71.84
143.87	143.33	72.12	135.00	0.81	0.80	0.01	0.75	90.00	1.20	71.37
143.33	142.76	71.58	135.00	0.80	0.80	0.01	0.82	90.00	1.20	70.76
142.76	142.26	71.10	135.00	0.80	0.79	0.01	0.72	90.00	1.20	70.38
142.26	141.81	70.67	135.00	0.79	0.79	0.00	0.65	90.00	1.20	70.02

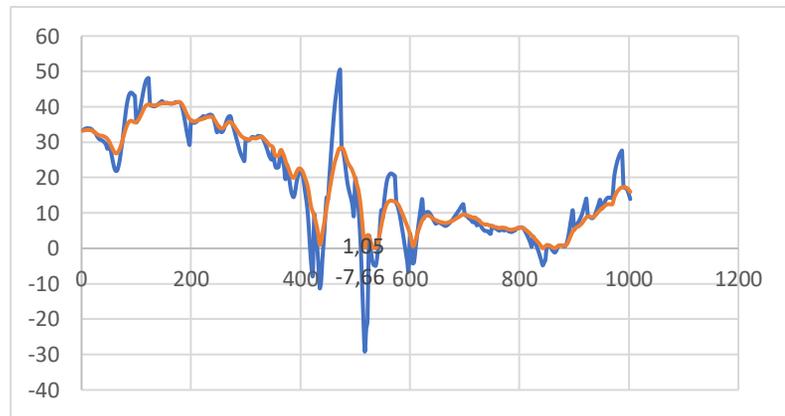
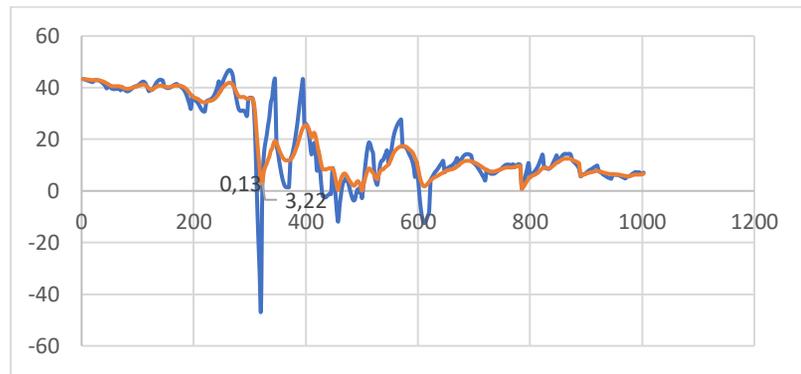
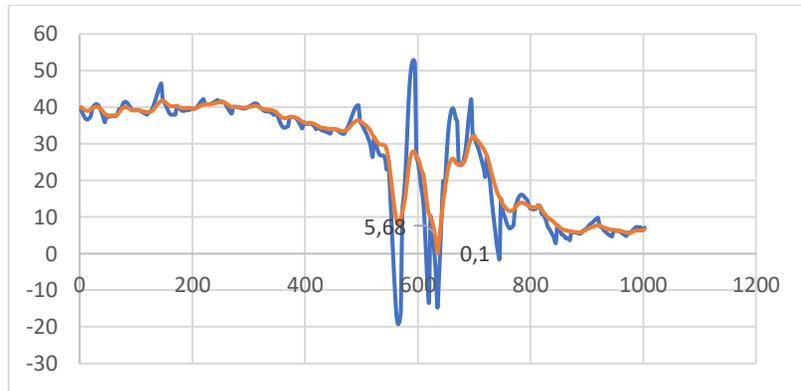
B. Datos registrados.

B.1 Datos obtenidos durante una caída sin flexionar las rodillas de aceleración y estado de la alarma.

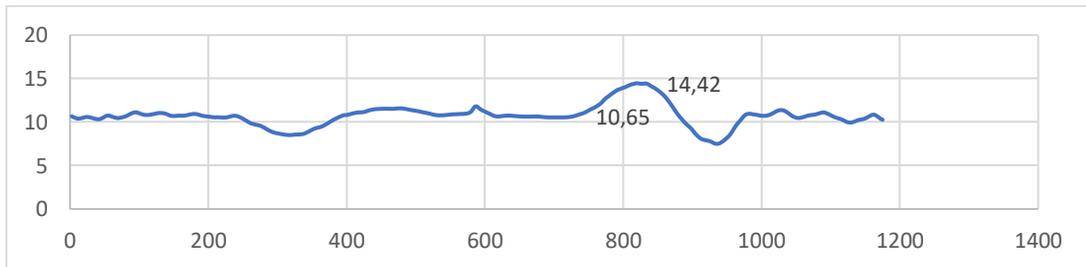
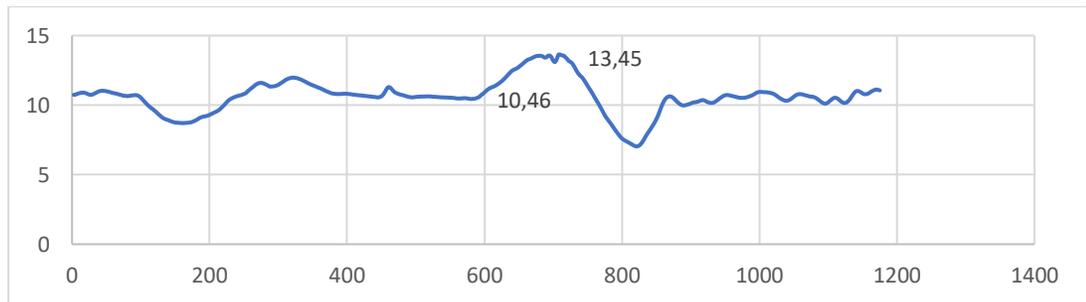
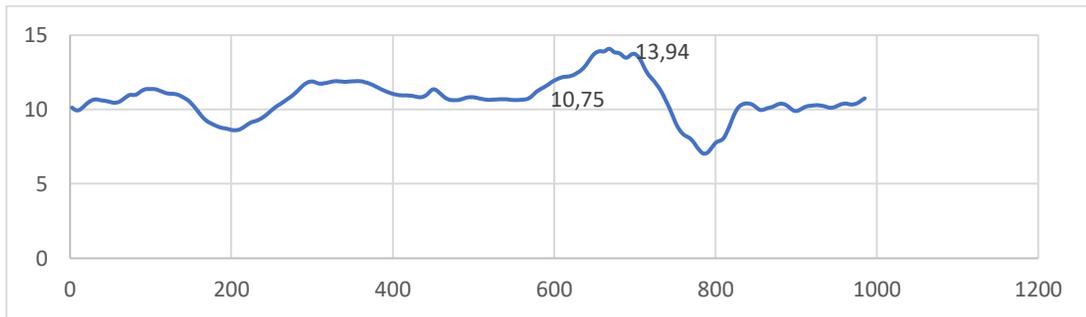
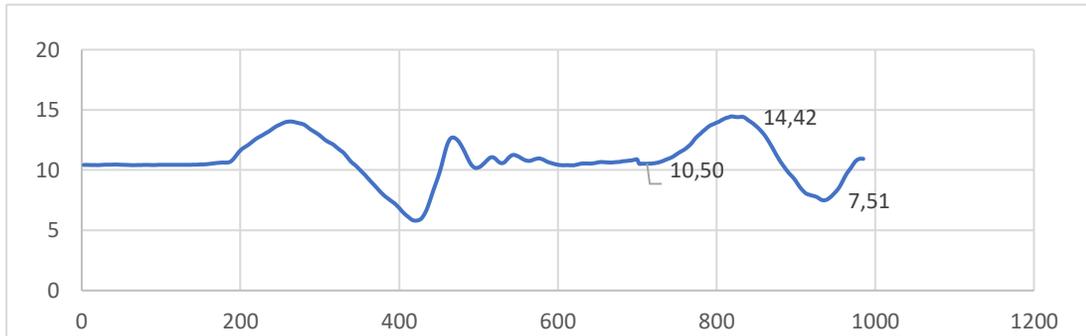


B.2 Datos obtenidos de distancia y desplazamiento angular de una actividad de caída flexionando las rodillas.

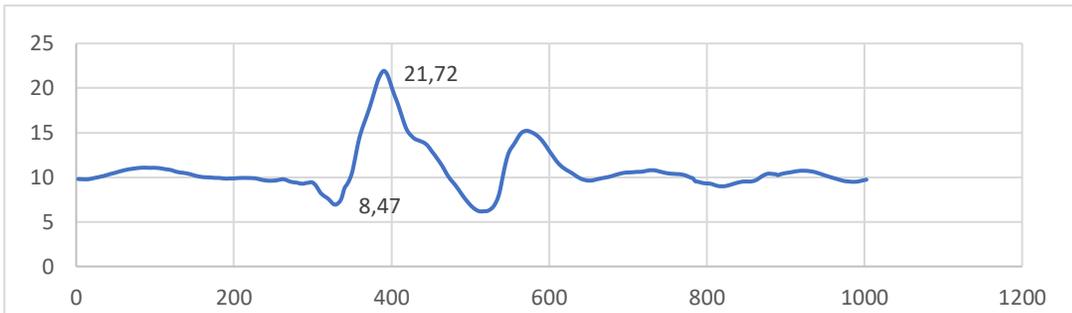
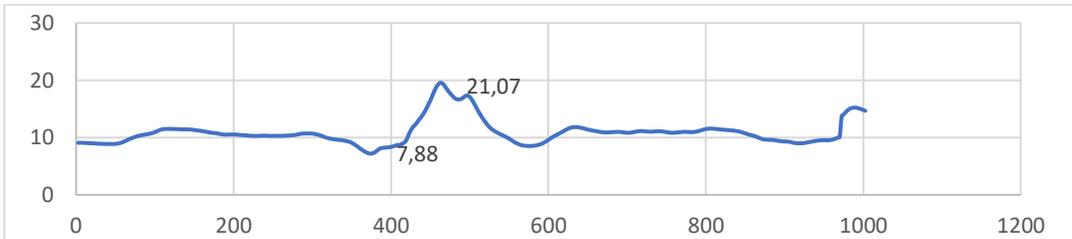
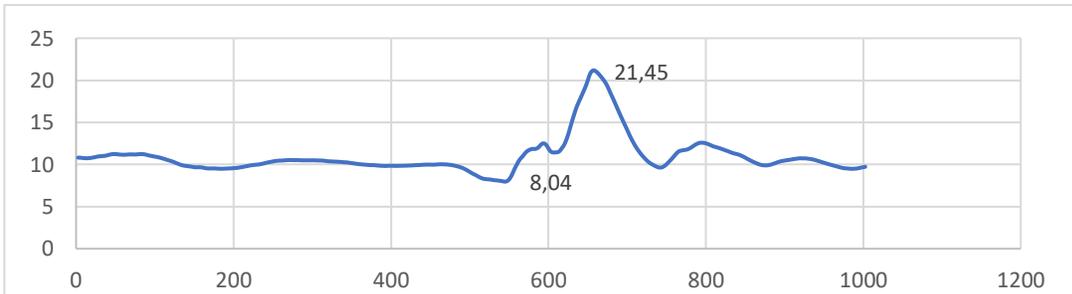
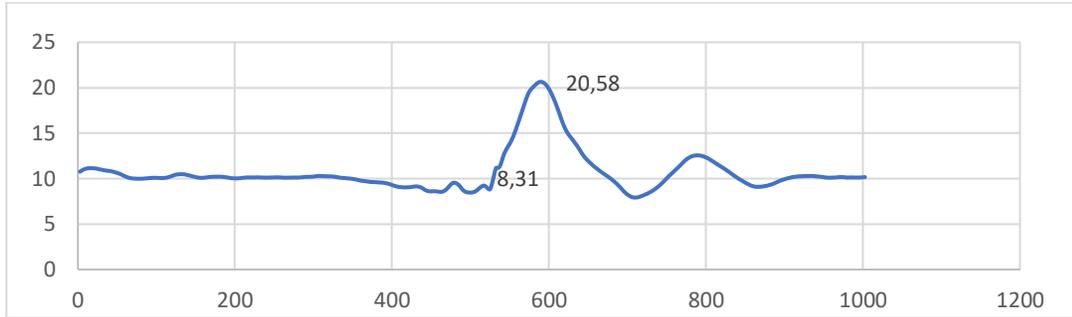




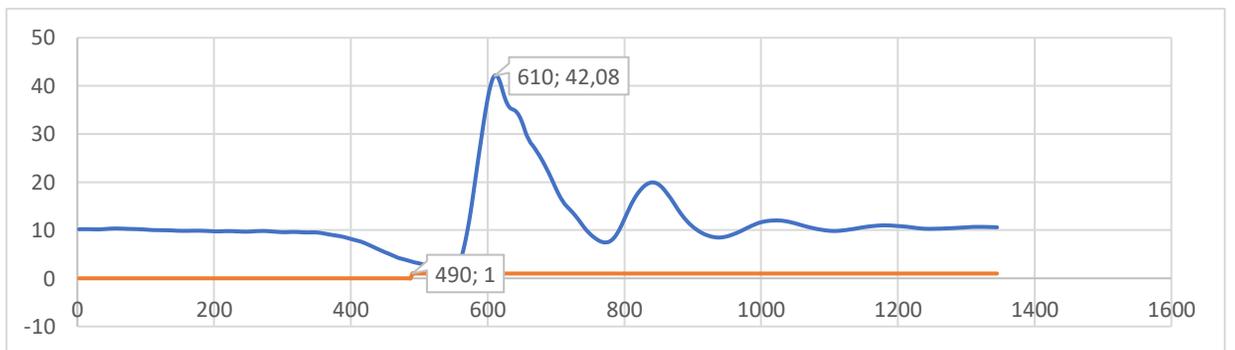
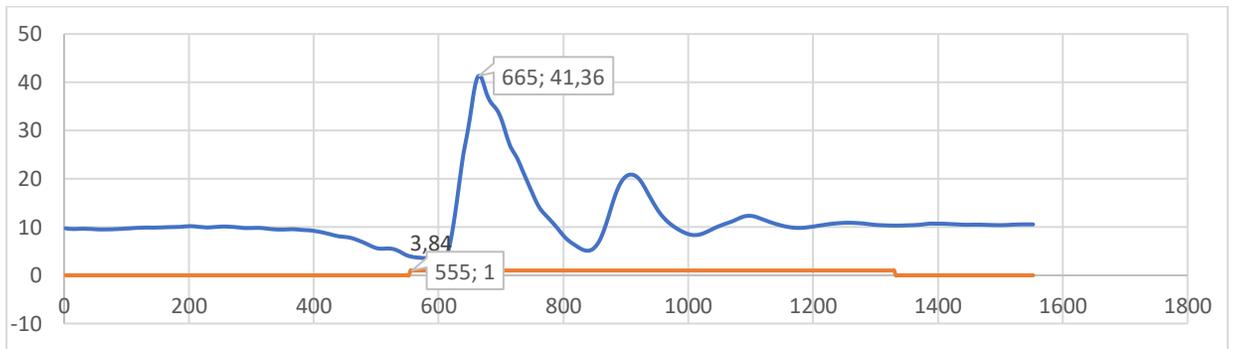
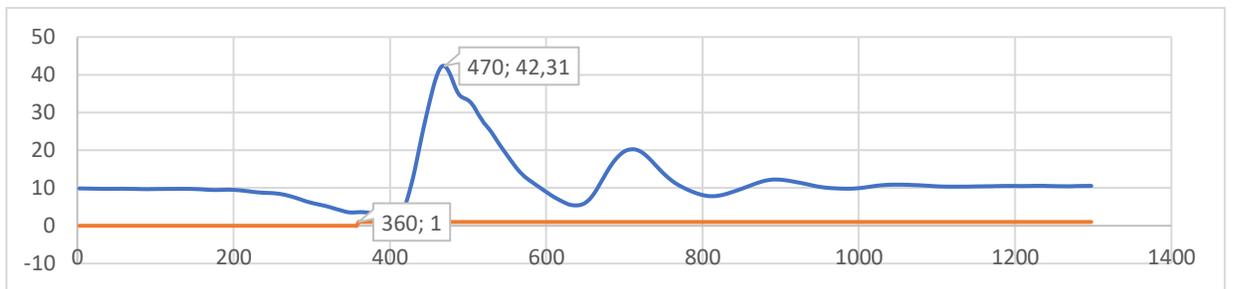
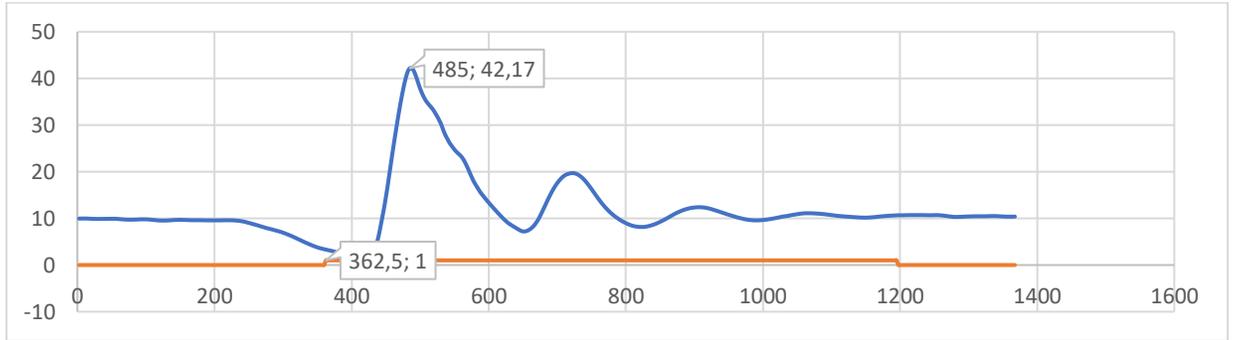
B.3 Datos obtenidos de aceleración en la rodilla de la actividad de agacharse.



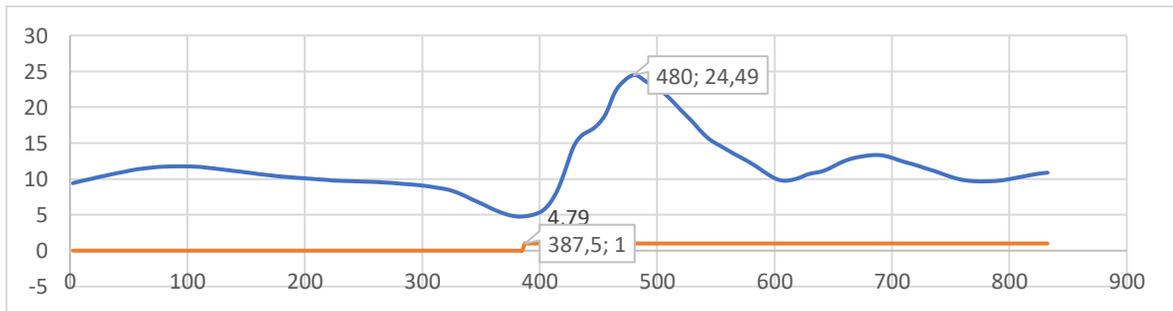
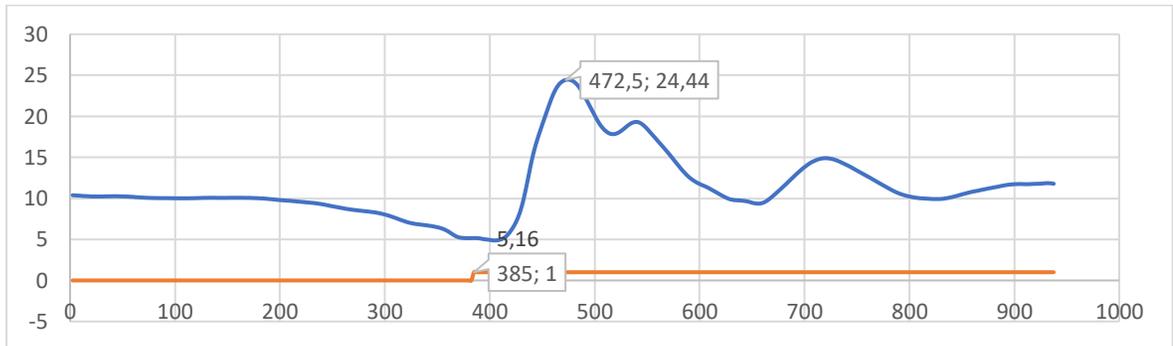
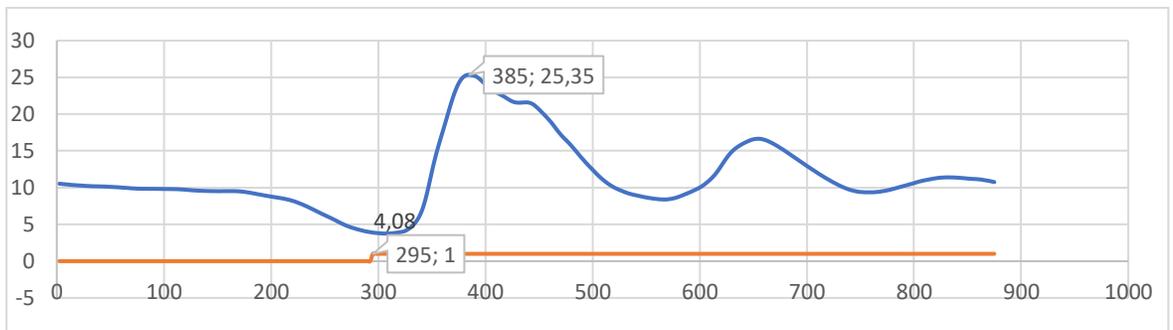
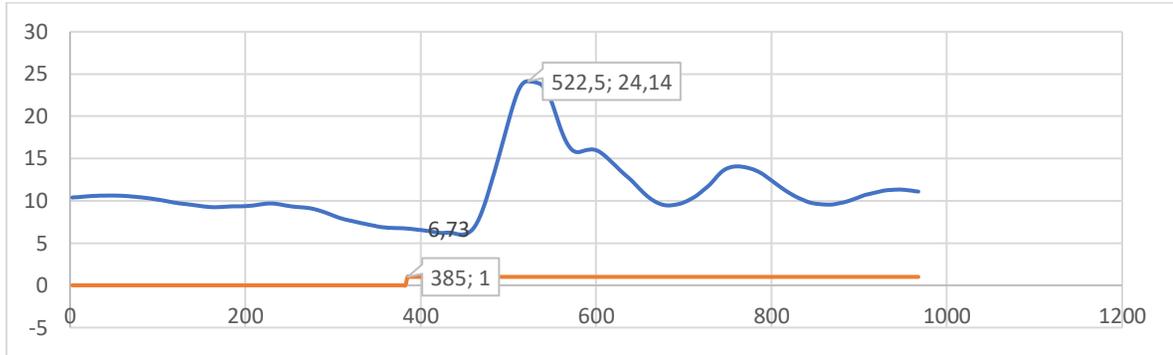
B.4 Datos obtenidos de aceleración en la rodilla de la actividad de caída flexionando las rodillas.



B.5 Datos obtenidos durante una caída sin flexionar las rodillas de aceleración y estado de la alarma.



B.6 Datos obtenidos durante una caída flexionando las rodillas de aceleración y estado de la alarma.



C. Codigos de programación.

C.1 Programa para el dispositivo uno ubicado en el centro de gravedad.

```
//librerias para los sensores
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
//librerias espnow
#include <esp_now.h>
#include <WiFi.h>
//configuracion del espnow
typedef struct struct_message {
    int id;
    int at1;
}struct_message;

// Crea funcion del mensaje
struct_message myData;

// Crea estructura para la tarjeta
struct_message board1;

// Crea un arreglo para la tarjeta
struct_message boardsStruct[2] = {board1};

// Funcion de recepcion de datos
void OnDataRecv(const uint8_t * mac_addr, const
uint8_t *incomingData, int len) {
    char macStr[18];
    //Serial.print("Packet received from: ");
    snprintf(macStr, sizeof(macStr),
"%02x:%02x:%02x:%02x:%02x:%02x",
        mac_addr[0], mac_addr[1], mac_addr[2],
mac_addr[3], mac_addr[4], mac_addr[5]);
    //Serial.println(macStr);
    memcpy(&myData, incomingData,
sizeof(myData));
    // Serial.printf("Board ID %u: %u bytes\n",
myData.id, len);
    // Update the structures with the new incoming data
boardsStruct[myData.id-1].at1 = myData.at1;
    //Serial.printf("x value: %d \n",
boardsStruct[myData.id-1].at1);
    //Serial.println();
}
//librerias para la conexion a internet y el servidor
smtp
#include <Arduino.h>
#include <ESP_Mail_Client.h>
//Ingresa las credenciales de la conexión wifi
#define WIFI_SSID "*****"
#define WIFI_PASSWORD "*****"

#define SMTP_HOST "smtp.gmail.com"
#define SMTP_PORT 465

#define AUTHOR_EMAIL "*****"
#define AUTHOR_PASSWORD "*****"

#define RECIPIENT_EMAIL
"barivadeneirap@utn.edu.ec"

//Cinematica
Adafruit_MPU6050 mpu;
float h=100; /*(cm)*/
float h2=1.2;
float betan_1=0;
float ang_y=0;

float ac_abs = 0;
float ac_absfi =0;
/*angulo giroscopio*/
```

```

long tiempo_prev;
float dt;
float ang_ygir;
float ang_y_prevgir;
//alarma
const int led1 = 4;
const int pininterruptor = 12;
bool estadoalarma;
bool umbral1;
int interruptor;
float sumatoria = 0 ;
int contador = 0;
void setup(void) {
  Serial.begin(115200); //115200
  pinMode(led1, OUTPUT);
  pinMode(pininterruptor, INPUT);
//esp now
//Se configura al dispositivo como estación WIFI
WiFi.mode(WIFI_STA);

//Init ESP-NOW
if (esp_now_init() != ESP_OK) {
  Serial.println("Error initializing ESP-NOW");
  return;
}
  esp_now_register_recv_cb(OnDataRecv);

  Serial.println();
  Serial.print("Connecting to AP");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(200);
  }
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");

```

```

Serial.println(WiFi.localIP());
Serial.println();
  while (!Serial)
    delay(10);
  if (!mpu.begin()) {
    while (1) {
      delay(10);
    }
  }

  mpu.setAccelerometerRange(MPU6050_RANGE_8_
G); /*Rangos del acelerometro
MPU6050_RANGE_2_G,MPU6050_RANGE_4_G,
MPU6050_RANGE_8_G,
MPU6050_RANGE_16_G */

  mpu.setFilterBandwidth(MPU6050_BAND_94_HZ);
/*Rangos de frecuencia 260-184-94-44-21-10-5*/

  mpu.setGyroRange(MPU6050_RANGE_1000_DEG)
; /*Rangos del giroscopio 250-500-1000-2000*/
  //delay(100);
}

void loop() {

  digitalWrite(led1, LOW);
  int board1Xat1 = boardsStruct[0].at1;

  sensors_event_t a, g, temp; /*
g.gyro.xyz,a.acceleration.xyz*/
  mpu.getEvent(&a, &g, &temp);

  betan_1 = ang_ygir;
  ac_abs = sqrt(pow(a.acceleration.x,2) +
pow(a.acceleration.y,2) + pow(a.acceleration.z,2));
  ac_absfi = (0.05*ac_abs)+((1-0.05)*ac_absfi);

```

```

    ang_y =
acos(a.acceleration.y/sqrt(pow(a.acceleration.x,2) +
pow(a.acceleration.y,2) +
pow(a.acceleration.z,2)))*(180.0/3.14);

    /*giroscopio*/
    dt = (millis() - tiempo_prev) / 1000.0;
    tiempo_prev = millis();
    /**/
    /*angulo*/
    ang_ygir = 0.90*(ang_y_prevgir +
(g.gyro.y/131)*dt) + 0.1*ang_y;
    ang_y_prevgir = ang_ygir;
    /**/
    float d =abs(h*cos((ang_ygir*3.14)/180));
    /*Ecuacion de trayectoria por pasos*/
    float s1=1.5*h;
    float s2= abs(cos((betan_1*3.14)/180));
    float s3= abs(cos((ang_ygir*3.14)/180));
    float s4 = s2-s3;
    float s5 = s1*s4;
    /*float s =1.5*h*(cos((betan_1*3.14)/180)-
cos((ang_yf*3.14)/180));*/

    contador = contador + 1;
    sumatoria = sumatoria + s5;
    if (contador == 25)
    {
        sumatoria = 0;
        contador = 0;
    }
    float resultado = d-sumatoria;
    //if (resultado<=h2 && ac_absfi <= 3.92)
    interruptor=digitalRead(pininterruptor);
    if (resultado<=h2)
    {
        umbral1 = 1;

        if(ac_absfi <=3.92 && umbral1==1){
            estadoalarma = 1;
        }
    }
    if (estadoalarma == 1 && interruptor == 0)
    {
        digitalWrite(led1, HIGH);
        //Serial.print("enviandocorreo");
        // correo();
    }
    if(ac_absfi <=3.92)
    {
    if (board1Xat1 == 1 && interruptor == 0)
    {
        digitalWrite(led1, HIGH);
        estadoalarma = 1;
        //Serial.print("enviandocorreo");
        correo();
    }
    }
    if (interruptor == 1){
        estadoalarma = 0;
        umbral1 = 0;
        digitalWrite(led1, LOW);
    }
    /*Probar las operaciones de la ecuacuacion de
trayectoria*/
    Serial.print(ac_absfi);
    Serial.print(",");
    Serial.print(ang_ygir);
    Serial.print(",");
    Serial.println(estadoalarma);
    //Serial.println("");
    delay(2.5);
}

void correo (){
    SMTPSession smtp;

```

```

smtp.debug(1);
ESP_Mail_Session session;
session.server.host_name = SMTP_HOST;
session.server.port = SMTP_PORT;
session.login.email = AUTHOR_EMAIL;
session.login.password =
AUTHOR_PASSWORD;
session.login.user_domain = "";
SMTP_Message message;
message.sender.name = "Nombre Apellido";
message.sender.email = AUTHOR_EMAIL;
message.subject = "Caída";
message.addRecipient("Familiar",
RECIPIENT_EMAIL);

```

```

String htmlMsg = "<div
style=\"color:#2f4468;\"><h1>Se ha detectado una
caída!</h1><p>- .....</p></div>";
message.html.content = htmlMsg.c_str();
message.html.content = htmlMsg.c_str();
message.text.charSet = "us-ascii";
message.html.transfer_encoding =
Content_Transfer_Encoding::enc_7bit;
if (!smtp.connect(&session)){
return;}
if (!MailClient.sendMail(&smtp, &message)){
Serial.println("Error sending Email, " +
smtp.errorReason());}
delay(10000);
}

```

C.2 Programa para el dispositivo dos ubicado en la rodilla.

```

#include <ESP8266WiFi.h>
#include <espnow.h>
//Libreiras Sensor
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
// Direccion MAC del ESP-32
uint8_t broadcastAddress[] = {0xEC, 0x94, 0xCB,
0x67, 0x14, 0xD8};

// Configurar el Id de la placa de desarrollo
#define BOARD_ID 1

// Estructura del mensaje de envio debe ser igual a la
estructa del receptor
typedef struct struct_message {
int id;
int at1;
} struct_message;
// Crear una funcion que guarde los datos para ser
enviados
struct_message myData;

unsigned long lastTime = 0;
unsigned long timerDelay = 2.5;

// Envia un mensaje de que se recibio el mensaje
void OnDataSent(uint8_t *mac_addr, uint8_t
sendStatus) {
Serial.print("\r\nLast Packet Send Status: ");
if (sendStatus == 0){

```

```

Serial.println("Delivery success");
}
else{
Serial.println("Delivery fail");
}
}
//Cinematica
Adafruit_MPU6050 mpu;

float ac_abs = 0;
float ac_absfi =0;

float h=44; /*(cm)*/
float h2=3.5;
float betan_1=0;
float ang_y=0;

/*angulo giroscopio*/
long tiempo_prev;
float dt;
float ang_ygir;
float ang_y_prevgir;

//alarma

const int led1 = 4;
const int pininterruptor = 12;
int estadoalarma = 0;
bool umbral1;

```

```

float sumatoria = 0 ;
int contador = 0;
int contador2 = 0;

void setup() {
  // Inicializa el monitor serie
  Serial.begin(115200);

  // Configura el dispositivo como estación wifi
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();

  // Inicializa el ESP-now
  if (esp_now_init() != 0) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }
  // Configura el role del esp now

  sp_now_set_self_role(ESP_NOW_ROLE_CONTROLLER);
  // Obtiene el estado del envio del mensaje
  esp_now_register_send_cb(OnDataSent);

  // Registro de pares
  esp_now_add_peer(broadcastAddress,
ESP_NOW_ROLE_SLAVE, 1, NULL, 0);

  Serial.println();

  while (!Serial)
    delay(10);

  if (!mpu.begin()) {
    while (1) {
      delay(10);
    }
  }
  mpu.setAccelerometerRange(MPU6050_RANGE_8_G); /*Configura el rango de
MPU6050_RANGE_2_G,MPU6050_RANGE_4_G,
MPU6050_RANGE_8_G,
MPU6050_RANGE_16_G */

  mpu.setFilterBandwidth(MPU6050_BAND_94_HZ);
/*Configura la frecuencia 260-184-94-44-21-10-5*/

  mpu.setGyroRange(MPU6050_RANGE_1000_DEG)
/*Configura el rango del giroscopio 250-500-1000-
2000*/
}

void loop() {

  sensors_event_t a, g, temp; /*
g.gyro.xyz,a.acceleration.xyz*/

```

```

mpu.getEvent(&a, &g, &temp);

ac_abs = sqrt(pow(a.acceleration.x,2) +
pow(a.acceleration.y,2) + pow(a.acceleration.z,2));
ac_absfi = (0.05*ac_abs)+((1-0.05)*ac_absfi);

betan_1 = ang_ygir;
ang_y =
acos(a.acceleration.y/sqrt(pow(a.acceleration.x,2) +
pow(a.acceleration.y,2) +
pow(a.acceleration.z,2)))*(180.0/3.14);

/*giroscopio*/
dt = (millis() - tiempo_prev) / 1000.0;
tiempo_prev = millis();
/**/
/*angulo*/
ang_ygir = 0.90*(ang_y_prevgir +
(g.gyro.y/131)*dt) + 0.1*ang_y;
ang_y_prevgir = ang_ygir;
/**/
float d =abs(h*cos((ang_ygir*3.14)/180));
/*Ecuacion de trayectoria por pasos*/
float s1=1.5*h;
float s2= abs(cos((betan_1*3.14)/180));
float s3= abs(cos((ang_ygir*3.14)/180));
float s4 = s2-s3;
float s5 = s1*s4;
/*float s =1.5*h*(cos((betan_1*3.14)/180)-
cos((ang_yf*3.14)/180));*/

contador = contador + 1;
contador2 = contador2 + 1;
sumatoria = sumatoria + s5;

if (contador == 5)
{
  sumatoria = 0;
  contador = 0;
}

float resultado = d-sumatoria;

if (resultado<=h2)
{
  umbral1 = 1;
  if(ac_absfi < 8.5 && umbral1==1){
    estadoalarma = 1;
  }
}

/* if (estadoalarma >= 1)
{
  Serial.print("enviandocorreo");
}

```

```

    // correo();
    }*/

if (contador2 == 10){
    estadoalarma = 0;
    contador2 = 0;
}

/*Probar las operaciones de la ecuacion de
trayectoria*/

/*Serial.print(resultado);
Serial.print(",");
Serial.print(ac_absfi);
Serial.print(",");
Serial.print(d);
Serial.print(",");
Serial.println(estadoalarma);*/

//Serial.println("");
delay(2.5);

if ((millis() - lastTime) > timerDelay) {
    // Set values to send
    myData.id = BOARD_ID;
    myData.at1 = estadoalarma;

    // Send message via ESP-NOW
    esp_now_send(0, (uint8_t *) &myData,
sizeof(myData));
    lastTime = millis();
}
}

```

C.3 Programa para determinar la dirección MAC del ESP-32.

```

#ifdef ESP32
    #include <WiFi.h>
#else
    #include <ESP8266WiFi.h>
#endif

void setup(){
    Serial.begin(115200);
    Serial.println();
    Serial.print("ESP Board MAC Address: ");
    Serial.println(WiFi.macAddress());
}

void loop(){
}

```