

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



TESIS DE GRADO

**PROTOTIPO PARA LA DETECCIÓN DE CAÍDAS EN
ADULTOS MAYORES MEDIANTE ARDUINO E IOT**

Tesis de Grado para obtener el Título de Licenciatura en Informática
Mención Ingeniería de Sistemas Informáticos

POSTULANTE: RUBÉN LIMACHI HUALLPA

TUTOR METODOLÓGICO: M. Sc. GROVER ALEX RODRIGUEZ RAMIREZ

ASESOR: Ph. D. YOHONI CUENCA SARZURI

LA PAZ – BOLIVIA

2020



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A mis queridos padres Roberto y Margarita quienes siempre estuvieron ahí a quienes me debo, por brindarme amor, comprensión, dedicación, consejos, fueron mi principal fuente de inspiración de superación de mi vida. Los quiero mucho.

A mis hermanos, Ruth y Raúl, por su apoyo, en todos los momentos más importantes de mi vida.

Esto es posible gracias a ustedes.

AGRADECIMIENTOS

A mi padre Roberto Limachi, mi madre Margarita Huallpa por brindarme amor incondicional, por preocuparse por mi bienestar, gracias por creer y confiar en mí, por apoyarme en mis estudios, por enseñarme los valores que me han llevado a alcanzar un gran meta en mi vida.

A mi hermana Ruth y mi hermano Raúl por haberme apoyado y brindado todos los consejos en los momentos más difíciles.

A mi tutor M. Sc. Grover Alex Rodriguez Ramirez por su colaboración, por el tiempo y paciencia que me brindo y me guio a lo largo de la revisión y corrección de la presente tesis.

A mi asesor Ph. D. Yohoni Cuenca Sarzuri, por el tiempo invertido, conocimiento, orientación, consejos y el asesoramiento brindado en este tiempo de la elaboración de la presente tesis.

A mis docentes de la carrera de informática por todo el conocimiento compartido que formaron parte de mi formación que a lo lardo de la estadía de la universidad.

A mis amigos que me brindaron su apoyo, quienes estuvieron ahí apoyándome, con quienes compartí grandes momentos, muchas experiencias, conocimientos, buenos y malos momentos, por brindarme su amistad y colaboración a lo largo de mi estadía en la universidad.

RESUMEN

El prototipo detector de caídas integra tecnologías como Arduino e Internet de las Cosas para detectar las caídas que sufren las personas adultas mayores con mayor frecuencia especialmente cuando viven solas, las caídas son una de las principales causas de mortalidad a nivel mundial, estas caídas pueden ser ocasionadas por diferentes causas entre ellas por enfermedades, pisos resbalosos, cables en el suelo, poca visibilidad, ambientes no adecuados u otros, las consecuencias de estas acaban en algunos momentos trágicas sin la atención apropiada y estos pueden alargar sus problemas en el futuro.

El presente trabajo de tesis pretende mostrar que es posible con la ayuda de tecnologías de Arduino e Internet de las Cosas detectar las caídas que sufren las personas adultas mayores, para poder responder a tiempo a una ayuda que requieran, cuando se detecte un evento de caída el prototipo advierte con una alarma sonora y la notificación de este evento a un familiar quien pueda tomar las medidas adecuadas.

El prototipo detector de caídas con respecto al hardware está compuesto por una placa de desarrollo NodeMCU que incluye un módulo wifi el cual facilita la conexión a internet, una unidad de medición inercial mpu-6050 quien está encargado de medir los cambios en las aceleraciones y velocidades angulares en los tres ejes x, y, z y determinar la orientación del cuerpo, un zumbador para las alarmas sonoras, un pulsador para cancelar las falsas alarmas en caso de que hubiera, así mismo empleando el lenguaje de programación C++ para la implementación del algoritmo. Los datos obtenidos se analizan a través de un algoritmo que observa la variación tanto del aceleración y velocidad angular y generar una alarma, en cuanto al software se utilizó IDE Arduino, Android Studio y un servicio de Firebase Cloud Messaging perteneciente a google, para que el sistema permita notificar a la aplicación móvil un evento de caídas.

Para el desarrollo del prototipo detector de caídas se utilizó la metodología en V, el cual se adecua a los requerimientos.

Palabras clave: mpu-6050, NodeMCU, Arduino, caídas, detección, Firebase, metodología V, IoT.

SUMMARY

The fall detector prototype integrates technologies such as Arduino and the Internet of Things to detect falls suffered by older adults more frequently, especially when they live alone, falls are one of the main causes of mortality worldwide, these falls can be Caused by different causes including illness, slippery floors, cables on the ground, poor visibility, unsuitable environments or others, the consequences of these end in some tragic moments without proper attention and these can lengthen your problems in the future.

This thesis work aims to show that it is possible with the help of Arduino and Internet of Things technologies to detect the falls suffered by the elderly, in order to respond in time to the help they require, when a fall event is detected. the prototype warns with an audible alarm and the notification of this event to a family member who can take the appropriate measures.

The hardware drop detector prototype is composed of a NodeMCU development board that includes a Wi-Fi module which facilitates internet connection, an inertial measurement unit mpu-6050 which is in charge of measuring changes in accelerations and speeds. angular in the three axes x, y, z and determine the orientation of the body, a buzzer for the audible alarms, a button to cancel the false alarms in case there were, likewise used the C ++ programming language for the implementation of the algorithm. The data obtained is analyzed through an algorithm that observes the variation of both the acceleration and angular velocity and generates an alarm. As for the software, IDE Arduino, Android Studio and a Firebase Cloud Messaging service belonging to google were used, so that the system allows to notify the mobile application of a crash event.

For the development of the fall detector prototype, the V methodology was used, which is adapted to the requirements.

Keywords: mpu-6050, NodeMCU, Arduino, caídas, detección, Firebase, V methodology, IoT.

ÍNDICE

CAPÍTULO I	1
1. MARCO REFERENCIAL.....	1
1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES	2
1.3. PLANTEAMIENTO DEL PROBLEMA	3
1.3.1. PROBLEMA CENTRAL	6
1.3.2. PROBLEMAS SECUNDARIOS.....	6
1.4. OBJETIVOS	6
1.4.1. OBJETIVO GENERAL.....	7
1.4.2. OBJETIVOS ESPECÍFICOS.....	7
1.5. HIPÓTESIS	7
1.5.1. OPERACIONALIZACIÓN DE VARIABLES	7
1.6. JUSTIFICACIÓN	7
1.6.1. JUSTIFICACIÓN SOCIAL.....	7
1.6.2. JUSTIFICACIÓN ECONÓMICA	8
1.6.3. JUSTIFICACIÓN CIENTÍFICA	8
1.7. ALCANCES Y LIMITES.....	8
1.7.1. ALCANCES.....	8
1.7.2. LIMITES	8
1.8. METODOLOGÍA.....	9
CAPÍTULO II	10
2. MARCO TEÓRICO	10
2.1. ARDUINO.....	10
2.2. RAZONES PARA UTILIZAR ARDUINO	11

2.3.	SENSORES ARDUINO	12
2.3.1.	SENSOR INERCIAL.....	12
2.3.2.	COMO FUNCIONA UN SENSOR INERCIAL	12
2.3.3.	ACELERÓMETRO	12
2.3.4.	GIROSCOPIO.....	14
2.3.5.	SENSOR IMU MPU-6050.....	16
2.3.6.	SENSOR IMU MPU-9250.....	17
2.3.7.	SENSOR MMA7361	17
2.3.8.	SENSOR ADXL325	18
2.3.9.	SENSOR ADXL335	19
2.3.10.	SENSOR ADXL345.....	19
2.3.11.	SENSOR BMG160.....	20
2.3.12.	SENSOR ITG-3200.....	20
2.4.	NODEMCU	21
2.5.	PLACA DE DESARROLLO NODEMCU	22
2.6.	INTERNET DE LAS COSAS IOT	22
2.7.	COMO FUNCIONA IOT	23
2.8.	APLICACIONES DE IOT	23
2.8.1.	APLICACIONES DEL IOT PARA EL SECTOR INDUSTRIAL	23
2.8.2.	APLICACIONES DEL IOT PARA EL USO DOMESTICO	24
2.8.3.	APLICACIONES DEL IOT EN CIUDADES INTELIGENTES.....	25
2.9.	PLATAFORMA IOT.....	25
2.10.	FIREBASE.....	26
2.11.	JSON	27
2.12.	NoSQL.....	28

2.13.	ANDROID STUDIO.....	29
2.14.	ADULTO MAYOR	30
2.15.	CAÍDA.....	30
2.15.1.	CLASIFICACIÓN DE LAS CAÍDAS.....	30
2.15.2.	ANÁLISIS DE UNA CAÍDA	30
2.16.	METODOLOGÍA V	31
2.17.	FRITZING	34
CAPÍTULO III.....		35
3.	DISEÑO METODOLÓGICO	35
3.1.	INTRODUCCIÓN.....	35
3.1.1.	PROCEDIMIENTOS DEL MODELO EN V.....	35
3.2.	FASE 1. ESPECIFICACIONES.....	36
3.2.1.	DIAGRAMA PARA EL MODELADO DE REQUISITOS.....	36
3.2.2.	DIAGRAMA DE CASOS DE USO.	36
3.3.3	DIAGRAMA DE ACTIVIDAD	39
3.3.	FASE 2. FUNCIONAL	39
3.4.	FASE 3. DISEÑO	40
3.4.1.	IDENTIFICACIÓN DE COMPONENTES DE HARDWARE	40
3.4.2.	IDENTIFICACIÓN DE COMPONENTES DE SOFTWARE.....	43
3.4.3.	DISEÑO DEL CIRCUITO ELECTRÓNICO.....	44
3.5.	FASE 4. CODIFICACIÓN	45
3.5.1.	SISTEMA ADQUISICIÓN DE DATOS.....	46
3.5.2.	APLICACIÓN MÓVIL	56
3.6.	FASE 5. TEST DE DISEÑO.....	59
3.7.	FASE 6. TEST FUNCIONAL.....	63

3.8. FASE 7. TEST DE ESPECIFICACIONES.....	66
CAPÍTULO IV	68
4. EVALUACIÓN DE RESULTADOS	68
4.1. INTRODUCCIÓN.....	68
4.2. PRUEBA DE HIPÓTESIS	68
4.3. ANÁLISIS DE RESULTADOS.....	71
CAPÍTULO V	72
5. CONCLUSIONES Y RECOMENDACIONES	72
5.1. INTRODUCCIÓN.....	72
5.2. CONCLUSIONES.....	72
5.3. RECOMENDACIONES.....	74
5.4. RECOMENDACIONES FUTURAS	75
BIBLIOGRAFÍA	76
ANEXOS	
DOCUMENTACIÓN	

ÍNDICE DE FIGURAS

Figura 2.1. Función del sensor inercial.....	13
Figura 2.2. Inclinación de un objeto en dos dimensiones.....	13
Figura 2.3. Inclinación en tres dimensiones	14
Figura 2.4. Funcionamiento del giroscopio	15
Figura 2.5. Sensor MPU-6050	16
Figura 2.6. Sensor MPU-9250	17
Figura 2.7. Sensor acelerómetro MMA7361	18
Figura 2.8. Sensor acelerómetro ADXL325	18
Figura 2.9. Sensor acelerómetro ADXL335	19
Figura 2.10. Sensor acelerómetro MMA7361	20
Figura 2.11. Sensor giroscopio BMG160.....	20
Figura 2.12. Sensor giroscopio ITG-3200	21
Figura 2.13. Placa de desarrollo NodeMCU.....	22
Figura 2.14. Servicios de Firebase.....	28
Figura 2.15. Entorno de desarrollo Android Studio	29
Figura 2.16. Etapas de la caída	31
Figura 2.17. Ciclo de vida de un proyecto de desarrollo de software	34
Figura 2.18. Logo de Fritzing	34
Figura 3.1. Diagrama casos de uso del prototipo detector de caídas	37
Figura 3.2. Diagrama de actividades del prototipo detector de caídas	39
Figura 3.3. Descripción general del sistema detector de caídas	40
Figura 3.4. Conexión de los componentes.....	44
Figura 3.5. Diagrama lógico de los componentes.....	45
Figura 3.6. Librerías utilizadas para el NodeMCU y el sensor.....	46

Figura 3.7. Código configuración de escala de Acelerómetro y giroscopio	47
Figura 3.8. Velocidad de transmisión	48
Figura 3.9. Aceleración durante una caída	48
Figura 3.10. Velocidad angular durante una caída	49
Figura 3.11. Código de magnitudes aceleración y velocidad angular	50
Figura 3.12. Magnitudes de caída frontal	50
Figura 3.13. Magnitudes al subir gradas.....	51
Figura 3.14. Magnitudes de caída lateral.....	51
Figura 3.15. Magnitudes de caída posterior.....	52
Figura 3.16. Magnitudes al caminar	52
Figura 3.17. Magnitudes de hacer saltos	52
Figura 3.18. Magnitudes al momento de sentarse en una silla	53
Figura 3.19. Aplicación de gravedad sobre un cuerpo	53
Figura 3.20. código de obtención de ángulos en Arduino	54
Figura 3.21. Diagrama de flujo para detectar caídas	55
Figura 3.22. Código para el envío de notificaciones	56
Figura 3.23. Nombre del proyecto en Consola de Firebase.....	57
Figura 3.24. Entorno de Android Studio	58
Figura 3.25. Configuración para añadir Firebase a la aplicación	58
Figura 3.26. Implementación de la clase en Android Studio.....	59
Figura 3.27. Código para obtención de token en Android Studio	59
Figura 3.28. Conexión física del prototipo	60
Figura 3.29. Conexión física del sensor MPU6050.....	60
Figura 3.30. Conexión física del zumbador.....	61
Figura 3.31. Conexión a Firebase y recepción de notificación en la aplicación.....	62

Figura 3.32. Prototipo final.....	63
Figura 3.33. prueba del prototipo	64
Figura 3.34. Magnitud de la aceleración y velocidad producida al caer de frente	65
Figura 3.35. Magnitud de la aceleración y velocidad al caer de espalda.....	65
Figura 3.36. Magnitud de aceleración y velocidad de una caída lateral derecho	65
Figura 3.37. Magnitud de aceleración y velocidad de una caída lateral izquierdo.....	66
Figura 3.38. Ubicación del prototipo y prototipo final	66
Figura 4.1. Aceptación de la hipótesis.....	71

ÍNDICE DE TABLAS

Tabla 3.1. Descripción casos de uso: colocar sistema	37
Tabla 3.2. Descripción de casos de uso: recolectar datos	38
Tabla 3.3. Descripción de casos de uso: determinar cálculo de caída	38
Tabla 3.4. Descripción de casos de uso: notificar caída	38
Tabla 3.5. Especificación del acelerómetro y giroscopio	41
Tabla 3.6. Especificación del módulo NodeMCU	41
Tabla 3.7. Especificación del Zumbador	42
Tabla 3.8. Especificación del pulsador	42
Tabla 3.9. Especificación del Protoboard	43
Tabla 3.10. Especificación del cable	43
Tabla 3.11. Estimación de inclinación del cuerpo en una caída	56
Tabla 3.12. Prueba de conexión al sensor MPU6050	61
Tabla 3.13. Prueba de conexión al sensor pulsador y zumbador	62
Tabla 3.14. Prueba de conexión al servidor	63
Tabla 3.15. Pruebas de caídas	64
Tabla 3.16. Personas de prueba	67
Tabla 3.17. Pruebas de caídas	67
Tabla 4.1. Valores observados de la detección de caídas	69
Tabla 4.2. Resultado de valor esperado	69
Tabla 4.3. Valores esperados	70
Tabla 5.1. Cantidad y costo total de materiales utilizados	73
Tabla 5.2. Detalle de costos de software	74

CAPÍTULO I

1. MARCO REFERENCIAL

1.1. INTRODUCCIÓN

El 'Internet of Things' o la internet de las cosas (IoT, por sus siglas en inglés) es un sistema de dispositivos de computación interrelacionados, máquinas mecánicas y digitales, objetos, lugares, animales, personas o cualquier otro objeto natural o artificial que tienen identificadores únicos y la capacidad de transferir datos a través de una red, sin requerir de interacciones humano a humano o humano a computadora. Aunque el concepto no fue nombrado hasta 1999, la internet de las cosas ha estado en desarrollo durante décadas (Rouse, 2017).

El Internet de las cosas hace referencia a la interconexión de los objetos del mundo físico a través de Internet y los cuales están equipados con sensores, actuadores y tecnología de comunicación (Tavizon, Guajardo, & Laines, 2016), el cual permite crear y promover un ambiente totalmente informado y confortable para una mejor toma de decisiones y calidad de vida (Mora, 2015).

Esta tecnología va encaminada hacia una gran variedad de ámbitos, tales como la industria, la salud y la energía, además de facilitar el desarrollo de nuevas aplicaciones y la mejora de las aplicaciones ya existentes (Tavizon, Guajardo, & Laines, 2016). Tanto así que se fueron creando una gran variedad de sensores, los cuales permiten adquirir registros de nuevos datos, variables entre otros, para su posterior análisis y toma de decisión.

Así como los sensores que tiene una gran variedad, también fueron avanzando los microprocesadores los cuales fueron creándose cada vez más pequeñas y asimismo de menor consumo, gracias a esto nos permite crear dispositivos hardware ligeros y pequeños que pueden tener aplicación con distintos fines y en distintas áreas, uno de los muchos ejemplos es la utilización de un módulo acelerómetro el cual permite detectar la caída de una persona entre otros.

Los avances tecnológicos permiten el monitoreo de las personas en diferentes situaciones, en Bolivia como en otros países para los próximos años, está prevista un aumento de la población adulta, según el INE (2017), dentro los próximos años la esperanza de vida

subirá a los 74 años de edad, por lo cual esta población se encuentra vulnerable a accidentes por lo cual es necesario el monitoreo ante cualquier tipo de incidentes que se presenten.

La propuesta del presente trabajo es realizar un dispositivo el cual ayudará a detección de caídas inesperadas de las personas adultas mayores en tiempo real, el cual permitirá actuar de manera autónoma mediante el uso de procesos que esté comprobado los valores detectados por los sensores, también permitirá al cuidador o familiar mantener informado sobre la situación de algún evento de caída que haya sufrido el adulto mayor.

1.2. ANTECEDENTES

El IoT no es algo nuevo, algunos de sus componentes existen desde hace décadas, el término Internet de las Cosas empezó a escucharse desde hace varios años; pero nunca como hoy se pudieron conjugar los avances que hacen de esta tecnología algo muy atractivo para quien la implementa. En especial, porque trae consigo nuevas herramientas para superar desafíos ambientales, de salud, movilidad, ventas, ahorros e innovadoras formas de generar negocios.

El IoT Básicamente se refiere a la conexión de dispositivos, se trata de objetos que mandan y reciben información por medio de sensores que indican su localización, temperatura o movimiento, entre otros datos. Estos objetos o cosas pueden ser coches, electrodomésticos e, incluso, monitores de corazón conectados a una red (Santiago, 2019).

Por otro lado, Arduino también fue evolucionando desde hace varios años, presentando una gran variedad de tarjetas electrónicas, su hardware está construido por un microcontrolador, Arduino también es una plataforma de desarrollo de Hardware libre, relativamente económica lo cual permite crear proyectos que se pueden comercializar (Estrada, 2017).

Así mismo también se crearon nuevos sensores con los cuales se capta más magnitudes físicas para tener un mejor control, con estas tecnologías es posible realizar prototipos de menor consumo energético.

El avance tecnológico permite el monitoreo de las personas en diferentes situaciones tales como las caídas, el cual es la segunda causa mundial de muerte por lesiones accidentales o no intencionales según la Organización Mundial de la Salud.

Para la presente investigación se procedió a consultar trabajos relacionados o similares realizados sobre el tema, obteniendo información de interés.

(Medrano, 2014) desarrollo un sistema novedoso basada en tecnología móvil el cual tuvo como objetivo detecta las caídas a través de un algoritmo mismo que fue implementado en el dispositivo móvil, la recopilación de datos fue realizada por los voluntarios que simularon las caídas y la información fue guardada en una base de datos para su posterior procesamiento.

(Padilla Roberto, 2015). El cual desarrollo un dispositivo capaz de monitorizar a una persona que se encuentre en peligro o en extravío, además de un medio de alerta para notificar en caso que la persona requiera de ayuda inmediata y los componentes utilizados en este proyecto son de muy bajo costo, y en comparación con otros dispositivos, es la monitorización satelital que realiza a las personas.

Por otra parte (Carnuccio, et al., 2017), desarrollaron un sistema prototipo basado en una placa Intel Galileo Generación I y el sensor acelerómetro el cual permite detectar la ocurrencia de una caída en una persona, además describen los tipos de actividades de la vida diaria que han tenido en cuenta para descartar falsos positivos en las caídas.

(Lanzarini & Ridolfi, 2017) los cuales implementaron un dispositivo detector de caídas basado en la plataforma CIAA de bajo costo, el cual es capaz de detectar la caída de una persona y dar alerta al centro de monitoreo como así también a los familiares, este dispositivo es de suma utilidad para personas de movilidad reducida.

(Vázquez, Martínez, Sánchez, & Sánchez, 2018) los cuales desarrollaron un detector de caídas el cual está basado en un acelerómetro para detectar los movimientos de los individuos y un Raspberry para procesar los datos, al detectar un movimiento drásticos este genera alarma remota utilizando wifi.

(González, Areces, & Rueda, 2018) los cuales diseñaron un sistema capaz de proporcionar a las personas mayores una vía rápida para pedir ayuda de forma remota, a la persona a cargo de su cuidado, utilizando la red celular. La funcionalidad principal es un botón de ayuda, a través del cual la persona a cargo del cuidado de ancianos recibirá un mensaje de texto cuando el anciano presione el botón. El sistema también tiene otras características de gran valor, como un grupo de alarmas automáticas, que alertan a través de SMS.

1.3. PLANTEAMIENTO DEL PROBLEMA

Según la Organización Mundial de la Salud (2014) en todo el mundo sigue aumentando la esperanza de vida de las personas mayores. En 2020, por primera vez en la historia, los

mayores de 60 años superarán en número a los menores de 5 años. Para 2050 se espera que la población mundial de más de 60 años llegue a los 2000 millones, un aumento considerable frente a los 841 millones de la actualidad. El 80% de estas personas mayores vivirán en países de ingresos bajos y medianos.

Lo cual significa que como sociedad tenemos el deber de ofrecerles una mejor calidad de vida a dichas personas que en su momento nos aportaron su trabajo y conocimiento.

En Bolivia, de acuerdo al Instituto Nacional de Estadística realizada en el censo del año 2012 la población adulta mayor represento el 8,1% del total de la población, al 2020 se tendrá un incremento del 9,5%, y al 2030 el 11,5% tendencia que refleja el inicio del proceso de envejecimiento de la población en el Estado Plurinacional de Bolivia, al recordarse el Día de las Personas de la Tercera Edad, instituido por Decreto Supremo N° 1421 y ratificado mediante Ley 369 como Día de la Dignidad de las Personas Adultas Mayores (INE, 2017).

Según Encuesta de Hogar 2016, señala que los adultos mayores que tienen seguro de salud público llegan al 60,9%, los que cuentan con el seguro de salud privado llegan al 1,0% y los que no tienen seguro al 38,0% (INE, 2017).

De acuerdo al informe emitido por el director del Instituto Nacional de Estadística Luis Pereira señala que La población de adultos mayores, con 60 o más años de edad, llegó a un millón de personas en Bolivia lo que representa más del 60% de la población de los cuales se concentran en el departamentos de La Paz con el 31%, Santa Cruz con el 16% y Cochabamba con el 16% (Garcia, 2017).

Como bien se observa la población adulta mayor va en crecimiento, el número de personas con edad superior a 65 años es cada vez mayor debido al aumento de esperanza de vida. Según la INE (2017), la esperanza de vida subirá a 70,5 años de edad para los hombres y 77,5 años para las mujeres para el 2020, cifras que ira en ascenso, por lo tanto, el modelo social actual varia con respecto a los datos anteriores.

La soledad en personas mayores estando sin compañía y sin atención, es algo habitual en la actualidad, más del 5% tiene un hogar unipersonal, alrededor de 608.645 personas que viven solas en Bolivia, casi la mitad que representa el 42.3% tiene más de 60 años, un porcentaje distribuido de manera similar entre hombres (122.591) y mujeres (134.359). En la familia el problema se produce, debido a que los miembros se encuentran trabajando, u otras

actividades razones que impiden que estos puedan ejercer de cuidadores a tiempo completo (Opinión , 2020).

En el contexto actual las personas al alcanzar a la edad adulta mayor sufren cambios físicos, cognitivos, funcionales, psicoemocionales, pérdida de las funciones musculares y elasticidad, existe una disminución en los reflejos de defensa y equilibrio, es decir cambios tanto internos como externos. Los cambios que el envejecimiento natural producen en el organismo humano afectan a la marcha, al equilibrio, al control neuromuscular, también sufren la disminución en la base de sustentación dejando así el centro de gravedad más inestable, los cuales constituyen las causas principales en la frecuencia de caídas de las personas adultas mayores, generando uno de los peligros que se enfrentan continuamente los adultos mayores (Camacho, 2019).

Todos estos cambios provoca en el deterioro de la salud, enfrentan enfermedades y en ocasiones controlarlas puede ser complicado debido a las diferentes patologías, así como evitar situaciones de riesgo como infartos, caídas inesperadas por tropiezos los cuales en su mayoría son accidentes que sufren las personas mayores en el hogar generalmente por una falta de adecuación de las instalaciones a sus condiciones físicas derivadas del envejecimiento: menor agudeza visual, deterioro de las capacidades físicas (marcha, coordinación, equilibrio), deterioro cognitivo y otros (Camacho, 2019).

Según la Organización Mundial de la Salud, las caídas están considerado como la segunda causa principal de muerte por lesiones accidentales o no intencionales, se calcula que anualmente mueren en todo el mundo unas 646000 personas debido a caídas, y más de un 80% de esas muertes se registran en países de bajos y medianos ingresos. “las caídas pueden ocasionar: fracturas, pérdida del conocimiento, hemorragia interna, lesiones en la piel, inflamaciones, dolor intenso en la zona del golpe y pérdida de funcionalidad” (Camacho, 2019).

Las caídas más frecuentes que sufren los adultos mayores durante sus actividades (levantarse, sentarse, subir gradas, caminar) son la caída frontal, posterior estos debido a resbalones, tropiezos, desmayos y caídas laterales los cuales son causadas al caer de una cama, una silla entre otros, en consecuencia al no contar con una forma eficaz de comunicarle a los familiares o personas que cuidan a los adultos mayores estos resultan gravemente afectados incluso llegando a perder la vida sin que nadie se percate de ello, no se cuentan con

herramientas tecnológicos que ayuden en situaciones de peligro como son las caídas de fallecimientos que se presentan, no hay una manera que permita informar o alertar al cuidador o sus familiares para recibir la atención y mejorar la calidad de vida de los adultos mayores entre otros.

Los adultos mayores no solo tienen mayor fragilidad y patologías de base mencionadas anteriormente, sino que también son los que tienen más dificultades para acceder a la tecnología que los mantenga en un contacto a distancia con sus familiares (Opinión, 2020).

Ante las actividades diarias de los familiares, se complica estar pendiente las 24 horas del día, por otro lado existen familias que tienen incorporados a los adultos mayores con todo el cuidado afectivo necesario, incluso la capacidad económica para pagar una institución que garantice una atención, por otro lado existen quienes no tienen la posibilidad de vivir en familia por no contar con una vivienda adecuada, presupuestos económicos bajos, la imposibilidad de pagar un centro geriátrico privado debido a que los costos que oscilan entre 1.500 y 7.000 bolivianos, lo que no les permite apoyar las necesidades de los adultos mayores (Opinión, 2020).

1.3.1. PROBLEMA CENTRAL

¿Cómo detectar las caídas laterales, posterior y frontal en los adultos mayores durante las actividades diarias para su pronta asistencia?

1.3.2. PROBLEMAS SECUNDARIOS

- Las caídas que sufre el adulto mayor causan daños físicos que agravan los problemas en la salud.
- EL desconocimiento de las caídas que sufren el adulto mayor por parte del responsable, genera una asistencia tardía.
- Entre mayor tiempo se encuentre inconsciente el adulto mayor debido a una caída, aumenta la probabilidad de muerte.
- Los responsables del adulto mayor no están siempre al pendiente de ellos, por lo cual no saben cuándo asistir ante una caída.
- Al contratar un servicio personalizado para el cuidado del adulto mayor, los responsables o familiares incurren en gastos económicos elevados.

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un prototipo portable mediante la plataforma Arduino e Internet de las Cosas para la detección de las caídas en adultos mayores.

1.4.2. OBJETIVOS ESPECÍFICOS

- Ayudar a los adultos mayores a minimizar los daños físicos, consecuencias en la salud causadas por una caída.
- Notificar al responsable del adulto mayor el evento de una caída, de tal manera disminuya el tiempo de asistencia.
- Minimizar la mortalidad ante un evento de caída mediante un aviso (alarma sonora e inalámbrica) en un determinado tiempo.
- Hacer uso de dispositivos electrónicos que van apegados al cuerpo y este dispositivo va ligado a una aplicación, el cual ofrece respuestas inmediatas, para acudir a la atención en caso de una caída.
- Hacer el uso de tecnologías libre para el desarrollo del prototipo.
- Realizar las pruebas de funcionamiento del prototipo

1.5. HIPÓTESIS

Mediante un prototipo para la detección de caídas basado en Arduino e Intente de las Cosas se permite detectar las caídas laterales, frontal y posterior en una persona adulto mayor, con un nivel de confianza del 95 %.

1.5.1. OPERACIONALIZACIÓN DE VARIABLES

- Variable independiente: Prototipo detector de caídas basado en Arduino e internet de las cosas.
- Variable dependiente: Caídas laterales, frontal y posterior.
- Variable interviniente: Personas adultas mayores.

1.6. JUSTIFICACIÓN

1.6.1. JUSTIFICACIÓN SOCIAL

El prototipo desarrollado estará enfocado al adulto mayor para mejorar la calidad de vida y a la vez ayudar a la persona a cargo ya sea un familiar o cuidador y tener un mayor control sobre ellos y así mismo las asistencias correspondientes, este dispositivo favorecerá ya que

permitirá reducir de alguna forma los efectos perjudiciales que producen las caídas en los adultos mayores entre los principales efectos están la muerte accidental y fracturas de huesos u otros. Por otro lado, también se puede lograr utilizar para otra determinada población, así como para las niñas, niños, personas con enfermedad de Parkinson, epilepsia entre otros, el dispositivo ayudara a la población en general.

1.6.2. JUSTIFICACIÓN ECONÓMICA

La presente investigación plantea un dispositivo desarrollado de forma económica accesible debido a la utilización de hardware y software libre para su funcionamiento y además es una forma de monitorizar y alertar a una persona sin gastar una fortuna.

1.6.3. JUSTIFICACIÓN CIENTÍFICA

El aporte al conocimiento, se aplica la plataforma del internet de las cosas y Arduino con el cual es posible explotar al máximo el potencial de los datos obtenidos, la combinación de estas dos tecnologías ayudaría a resolver no solo un problema en la sociedad, por lo tanto, aportara información y conocimiento para su posterior tratamiento ya que con el paso del tiempo podrían desarrollarse mejoras tecnológicas.

1.7. ALCANCES Y LIMITES

1.7.1. ALCANCES

El prototipo tendrá las siguientes funcionalidades:

- El prototipo detecta las caídas frontal, posterior y laterales de una persona.
- El prototipo cuenta con alarma sonora e inalámbrica.
- El prototipo enviara alerta en caso de que la personas haya sufrido una caída.
- Él envío de alerta ante un evento de caída es automática.
- El prototipo utilizará un microcontrolador NodeMCU, utilizará una batería, sensores, para él envío de alertas al responsable o familiar.
- El funcionamiento es dentro del hogar.

1.7.2. LIMITES

El presente trabajo se limitará a los siguientes aspectos:

- EL prototipo solo envía alertas de caídas.
- EL prototipo no hace el monitoreo de la temperatura corporal, frecuencia cardiaca.
- No enviara notificación de alertas de caída en caso de no tener conexión a internet.

- El prototipo no podrá ser utilizados en lugares externos del hogar.

1.8. METODOLOGÍA

En el presente trabajo se utilizará la metodología en V, es un proceso que representa la secuencia de paso en el desarrollo del ciclo de vida de un proyecto. Su simbología se debe a que la letra V representa las palabras verificación y validación, sirve para indicar en qué fase de desarrollo se deben definir las pruebas correspondientes y también sirve para saber a qué fase de desarrollo hay que volver si se encuentra fallos en las pruebas correspondientes.

El modelo en v se divide en dos áreas principales, la parte izquierda en donde se realizan los requisitos el desarrollo general hasta llegar a la codificación o programación y la derecha que comprende la fase de validación.

El modelo está pensado para saber dónde se debe iniciar si una de las pruebas falla, volver a una de las fases determinada donde se encuentra la falla y recorrer todas y cada una de las fases, así sucesivamente hasta no volver a encontrar errores para así tener el prototipo final funcionando.

El primer nivel del modelo en V corresponde a la etapa o fase de especificaciones donde se define los requisitos del sistema, el segundo nivel la fase de diseño donde definimos el diseño global del sistema y por último la fase de codificación ya que realizara el hardware con los esquemas de la placa de desarrollo para su posterior integración con el internet y finalmente se realizarán las pruebas.

Una de las ventajas de la metodología en V, es un modelo sencillo y fácil de aprender, la relación entre las etapas de desarrollo y los distintos tipos de prueba facilitan la localización de fallos, involucra al usuario en la prueba a realizar.

2. MARCO TEÓRICO

2.1. ARDUINO

Es una plataforma electrónica de código abierto u "open-source", el cual permite realizar proyectos interactivos de manera sencilla. Está compuesta respectivamente por circuitos impresos que integran un microcontrolador y un entorno de desarrollo (IDE), en donde se programa cada placa, por tanto Arduino es tanto Software como Hardware (Raigón , 2017).

Arduino es una plataforma basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing. Es un dispositivo que conecta el mundo físico con el mundo virtual, o el mundo analógico con el digital (Estrada s. , 2014).

Arduino también simplifica el proceso de trabajar con microcontroladores, pero ofrece algunas ventajas para profesores, estudiantes y aficionados interesados sobre otros sistemas:

- **Barato:** las placas Arduino son relativamente baratas en comparación con otras plataformas de microcontroladores. La versión menos costosa del módulo Arduino se puede ensamblar a mano.
- **Multiplataforma:** El Software Arduino (IDE) se ejecuta en Windows, Macintosh OS X, y Linux. La mayoría de los sistemas de microcontroladores están limitados solo a Windows.
- **Entorno de programación simple y claro:** El software Arduino (IDE) es fácil de usar para principiantes, pero lo suficientemente flexible como para que los usuarios avanzados también lo aprovechen. Para los maestros, está convenientemente basado en el entorno de programación de procesamiento, por lo que los estudiantes que aprendan a programar en ese entorno estarán familiarizados con el funcionamiento del IDE de Arduino.
- **Código abierto y hardware extensible:** Los planes de las placas Arduino se publican bajo una licencia Creative Commons, por lo que los diseñadores de circuitos experimentados pueden hacer su propia versión del módulo, extenderlo y mejorarlo. Incluso los usuarios relativamente inexpertos pueden construir la versión del módulo de prueba para comprender cómo funciona y ahorrar dinero (Arduino, 2020).

2.2. RAZONES PARA UTILIZAR ARDUINO

Según Veloso (2016), el cual indica que existen muchas razones validas para utilizar Arduino, entre ellas se describe a continuacion.

- **Es Open Source:** Esto quiere decir que se puede acceder a toda parte del circuito y del código de las librerías, puedes modificarlas y no necesitas ninguna licencia para utilizarlo.
- **Facil de programar:** Arduino cuenta con una infinidad de librerías que nos resuelven la vida, gran porcentaje del código mas complejo que necesitamos como por ejemplo para establecer una comunicación serie o inicializar un sensor de temperatura ya están resueltos en librerías, solo tenemos que usarlas. Otra gran ventaja es que no se necesita ninguna tarjeta de programación como sucede en la mayoría de las placas de desarrollo, Arduino cuenta con un software conocido como bootloader que viene cargado en el microprocesador que se auto programa a si mismo y nos evita la necesidad de contar con una tarjeta programadora para programar el micro.
- **Documentación:** Al ser de Hardware y Software libre hay infinidad de información disponible dando vueltas por la red.
- **Amplia variedad de placas:** Arduino no es una única placa, es toda una familia de placas, cada una de ellas cuenta con diferencias y similitudes, según el proyecto que tengamos que hacer utilizamos una u otra.
- **Costo:** Podemos conseguir nuestra placa Arduino por menos de 14 dolares, es un precio muy económico comparado con otras placas que intentan cumplir los mismos requisitos. Obviamente existen diferentes modelos y alternativas, el costo puede varias pero no demasiado.
- **Diversas aplicaciones:** Podemos construir una infinidad de aplicaciones a bajo costo y con poco tiempo de desarrollo, si tenemos imaginación podemos hacer casi cualquier cosa.
- **Proyectos profesionales:** Si la tarjeta de Arduino cuenta con el microcontrolador en un zócalo desmontable como por ejemplo el Arduino uno. Podemos utilizar la placa solo para programarlo y dado el caso de tener que hacer algo mas profesional, podemos sacar el micro de la placa y utilizarlo sin el board, claro que tendríamos que cablear

todas las entradas/salidas y las fuentes de alimentación junto con la circuitería externa de nuestro proyecto.

2.3. SENSORES ARDUINO

Según Crespo (2019) el cual indica que un sensor es un dispositivo que permite conocer, capaz de detectar distintas magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas.

Las variables de instrumentación pueden ser, por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc.

Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD¹), una capacidad eléctrica (como en un sensor de humedad o un sensor capacitivo), una tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), otros.

2.3.1. SENSOR INERCIAL

Un sensor inercial o también conocido como IMU² es un componente capaz de obtener la posición, orientación y velocidad de cualquier dispositivo donde sea utilizado para captura y análisis de movimientos. Su construcción puede ser un poco compleja debido a que está compuesto por 3 diferentes sensores. Dentro de este se incorporan giroscopios, acelerómetros y magnetómetros (Mecafenix, 2018), la cual está basada en tecnologías MEMS con características más ligeras y más pequeñas que las habituales. Esta sirve para producir una medición en tres dimensiones de la fuerza específica y la velocidad angular.

2.3.2. COMO FUNCIONA UN SENSOR INERCIAL

Según Mecafenix (2018) cada uno de los sensores agregados aporta una función para lograr un único resultado. El giroscopio se encarga de medir los giros realizados, mientras que el acelerómetro mide la aceleración lineal que se realiza hacia cualquier lado y por último el magnetómetro obtiene información acerca del norte magnético para siempre estar ubicado con respecto al campo magnético de la tierra, como se muestra en la figura 2.1, Para este estudio solo se hará referencia al acelerómetro y giroscopio.

2.3.3. ACELERÓMETRO

¹ Siglas de resistance temperature detector

² Siglas de unidad de medición inercial

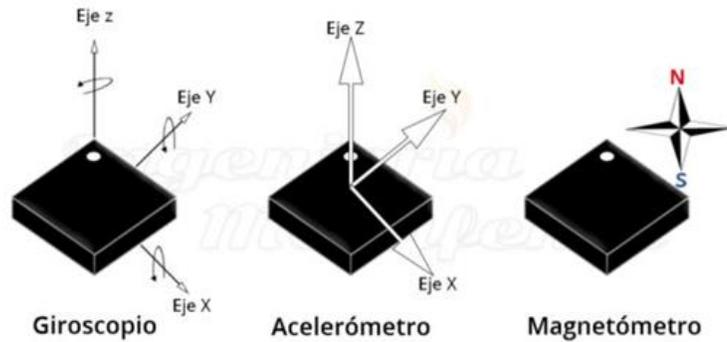


Figura 2.1. Función del sensor inercial

Fuente: (Mecafenix, 2018)

Un acelerómetro es un dispositivo electromecánico utilizado para medir las fuerzas de aceleración. Dichas fuerzas pueden ser estáticas, como la fuerza de gravedad continua o, como es el caso con muchos dispositivos móviles, dinámicas para detectar movimientos o vibraciones (Goodrich, 2013).

La aceleración puede expresarse en tres ejes X, Y, Z los cuales son las tres dimensiones del espacio, lo que permite saber simultáneamente la magnitud y dirección de la aceleración medida, así también la IMU puede detectar la aceleración de la gravedad el cual es 9.8 m/s^2 y gracias a la gravedad terrestre se puede usar las lecturas del acelerómetro para saber cuál es el ángulo de inclinación respecto al eje X, eje Y o al eje Z (Tr4nsduc7or, 2014).

Para determinar la orientación del sensor se obtiene usando trigonometría lo cual se observa en la figura 2.2, en el caso de 2D, asumiendo que ponemos el sensor horizontal en su plano X-Y, con Z apuntando hacia arriba, y únicamente giramos en el eje Y, la ecuación para el ángulo resulta lo siguiente.

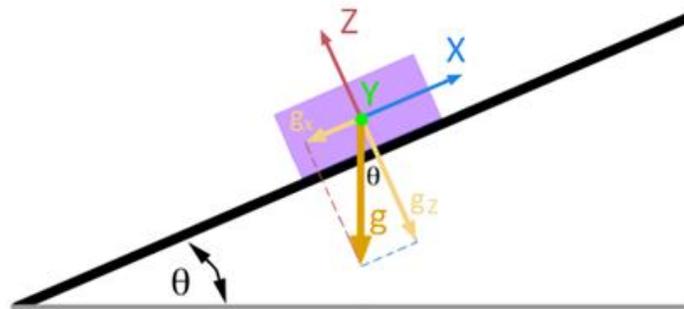


Figura 2.2. Inclinación de un objeto en dos dimensiones

Fuente: (Llamas, 2016)

$$\theta = \text{atan} \frac{A_x}{A_z}$$

De forma similar, en el caso 3D las ecuaciones resultan, lo cual se observa en la figura 2.3,

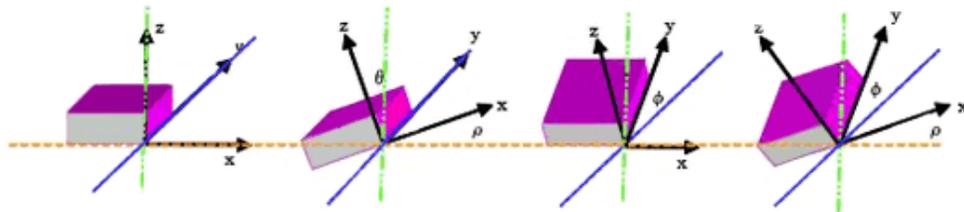


Figura 2.3. Inclinación en tres dimensiones

Fuente:(Llamas, 2016)

$$\theta_x = \text{atan} \frac{a_z}{\sqrt{a_y^2 + a_z^2}}$$

$$\theta_y = \text{atan} \frac{a_y}{\sqrt{a_x^2 + a_z^2}}$$

$$\theta_z = \text{atan} \frac{\sqrt{a_x^2 + a_y^2}}{a_z}$$

Un acelerómetro es un dispositivo muy versátil, que permite obtener mediciones de aceleraciones y orientación. Además, son sensores absolutos, es decir, que existe un punto 0,0,0 de medición, que correspondería con el sensor cayendo libremente en el vacío (Llamas, 2016).

2.3.4. GIROSCOPIO

Un giroscopio es un dispositivo que utiliza la gravedad de la Tierra para ayudar a determinar la orientación. Su diseño consiste en un disco de rotación libre llamado rotor, montado en un eje giratorio en el centro de una rueda más grande y más estable. A medida que el eje gira, el rotor permanece estacionario para indicar el tirón gravitacional central y, por lo tanto, en qué dirección está "hacia abajo" (Goodrich, 2018).

Según Llamas (2016) el cual indica, los giroscopios son dispositivos puramente diferenciales, es decir, no existe una referencia absoluta si no que siempre se mide ángulos relativos a una referencia arbitraria.

Existen diversos tipos de giroscopios (mecánicos, de anillo láser, de fibra óptica). Los que se emplean en MEMS son denominados giroscopios vibratorios de efecto Coriolis (CVG), que en cierta forma pueden considerarse una evolución del péndulo de Foucault.

El principio de funcionamiento es un CVG es que un objeto vibratorio tiene a vibrar en el mismo plano incluso si este rota. El efecto Coriolis provoca que el objeto vibratorio ejerza una fuerza sobre el soporta, y midiendo esta fuerza podemos determinar la rotación a la que está sometido el giroscopio la cual se ilustra en la figura 2.4.

Para registrar el efecto de la fuerza Coriolis un MEMS dispone de estructuras similares a las del acelerómetro. Ciertas partes del cuerpo se someten a vibración por resonancia y el efecto de la fuerza de Coriolis deforma la estructura, lo cuál puede ser medido por la variación de la capacitancia del sistema.

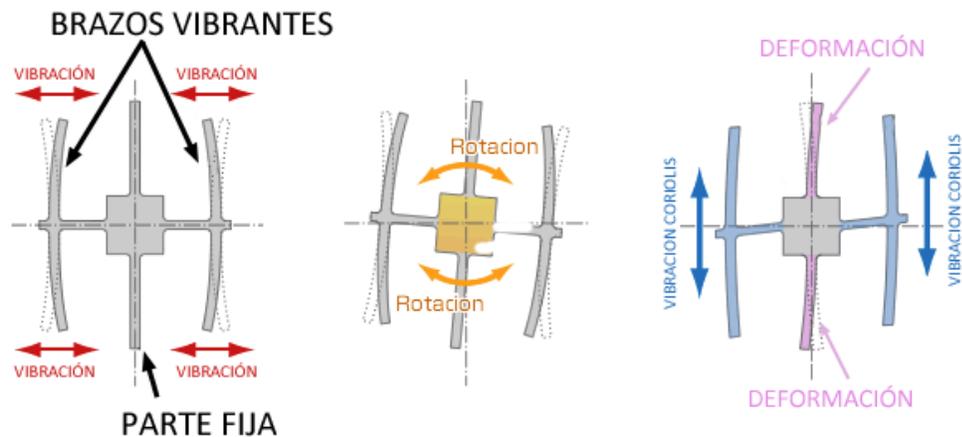


Figura 2.4. Funcionamiento del giroscopio

Fuente: (Llamas,2016)

Al igual que en el caso de los acelerómetros, lo normal es que los giroscopios que empleemos sean de 3 ejes, es decir, que registran de forma independiente la rotación en X, Y, y Z, lo cual permite determinar la magnitud y dirección de la rotación.

Una de las consecuencias de emplear la fuerza Coriolis es que los giroscopios vibratorios, a diferencia de otros tipos de giroscopios, no registran el ángulo girado si no la velocidad angular, que es la relación de variación del ángulo respecto del tiempo.

2.3.5. SENSOR IMU MPU-6050

EL módulo MPU6050 contiene un giroscopio de tres ejes con el que podemos medir velocidad angular y un acelerómetro también de 3 ejes con el que medimos los componentes X, Y y Z de la aceleración, el acelerómetro trabaja sobre el principio piezo eléctrico, posee además de un sensor de temperatura.

Los acelerómetros internamente tienen un MEMS (MicroElectroMechanical Systems) que de forma similar a un sistema masa-resorte permite medir la aceleración. Los giroscopios utilizan un MEMS para medir la velocidad angular usando el efecto Coriolis.

El MPU6050 es un sensor de movimiento, que tiene un conversor ADC de 16 bits que convierte los datos a un valor digital, el módulo de giroscopio se comunica con el Arduino a través de la comunicación serie I2C a través del reloj serial (SCL) y datos (SDA), el chip MPU6050 necesita 3.3V, pero un regulador de voltaje en la tarjeta GY-521 le permite alimentarlo hasta 5V.

El procesador interno del IMU es capaz de realizar cálculos precisos de los valores que miden sus sensores internos que son, aceleraciones lineales y angulares, para informarnos de valores útiles como los ángulos de inclinación con respecto a los 3 ejes principales (García, 2018).

La figura 2.5 se observa la ilustración del sensor MPU-6050 en la cual se observan los pines de conexión vcc, gnd, scl, sda, xda, xcl, add, int, las direcciones del eje x, y, z así mismo los microcomponentes por las cuales está compuesto el sensor.



Figura 2.5. Sensor MPU-6050

Fuente: (Llamas 2016)

2.3.6. SENSOR IMU MPU-9250

El MPU9250 combina un acelerómetro de 3 ejes, giroscopio de 3 ejes y magnetómetro de 3 ejes con un procesador de movimiento digital (DMP) integrado. La información que podemos obtener de un MPU9250 es: ángulo de guiñada, ángulo de cabeceo y ángulo de balanceo.

La MPU tiene un registro de 16 bits para cada uno de sus tres sensores. Almacenan temporalmente los datos del sensor antes de que se transmitan a través de I2C o SPI, se alimenta de 3.3 V hasta 5 V.

El MPU-9250 es una opción de acelerómetro, giroscopio y magnetómetro de alto rendimiento para su uso en aplicaciones como el reconocimiento de gestos en juegos, robots de auto equilibrio, juguetes, teléfonos celulares, navegación de vehículos, monitoreo de condición física y aplicaciones similares donde se detecta la dirección y magnitud del movimiento. junto con la rotación se desea (Shubham & Samhita , 2016).

La figura 2.6 se observa la ilustración del sensor MPU-9250.



Figura 2.6. Sensor MPU-9250

Fuente: (Shubham & Samhita 2016)

2.3.7. SENSOR MMA7361

Este sensor MMA7361 es un acelerómetro analógico de 3 ejes (x,y,z). El nivel de las medidas del acelerómetro, permite medir la aceleración, o la inclinación de una plataforma con respecto al eje terrestre.

Otras características que tiene son el modo sleep, acondicionamiento de señal, filtro pasa bajas de 1 polo, compensación de temperatura, auto prueba, y detección de 0g para caída libre.

Este sensor funciona con alimentaciones entre 2.2 y 3.6VDC (3.3V es el valor óptimo), y consume solamente 400 μ A de corriente (Haro, 2015).

La figura 2.7 se observa la ilustración del sensor MMA7361.

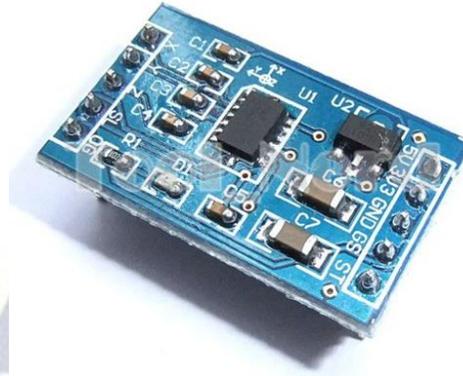


Figura 2.7. Sensor acelerómetro MMA7361

Fuente: (Shubham & Samhita 2016)

2.3.8. SENSOR ADXL325

El ADXL325 es un acelerómetro completo de 3 ejes pequeño, de baja potencia con salidas de tensión condicionadas por señal, el sensor mide aceleración con un rango mínimo de escala completa de ± 5 g. Puede medir la aceleración estática de la gravedad en aplicaciones de detección de inclinación, así como la aceleración dinámica, resultante del movimiento, choque, o vibración (Analog Devices, 2013).

En la figura 2.8 se observa la ilustración del sensor ADXL325.



Figura 2.8. Sensor acelerómetro ADXL325

Fuente:(<https://es.rs-online.com/web/p/herramientas-de-desarrollo-de-sensores/8031563/>)

2.3.9. SENSOR ADXL335

Este sensor ADXL335 es un acelerómetro analógico completo de 3 ejes X, Y Z con señales reguladas de voltaje de salida analógicas. Mide la aceleración con una escala de $\pm 3G$ y utiliza un nivel de tensión de 3.3 V a 5 V. Se puede medir la aceleración de la gravedad estática en aplicaciones de detección de inclinación, así como la aceleración dinámica resultante del movimiento, choque o vibraciones (Dub, 2015).

En la figura 2.9 se observa la ilustración del sensor ADXL335.



Figura 2.9. Sensor acelerómetro ADXL335

Fuente: (Dub,2015)

2.3.10. SENSOR ADXL345

Es un acelerómetro de 3 ejes X, Y, Z, de alta resolución que puede medir hasta $\pm 16 g$. Es pequeño y delgado, y tiene bajo consumo de energía. Los datos se pueden obtener a través de interfaz SPI de 3 o 4 alambre, o I2C (Volt, 2016).

El ADXL345 es adecuado para medir la aceleración estática de la gravedad para aplicaciones de detección de inclinación del sensor, gracias a la acción de la fuerza de la gravedad, así como la aceleración dinámica ocasionada por algún movimiento o impacto. Su alta resolución (4 mg/LSB) permite medir cambios de inclinación de menos de 1.0.

La tensión de alimentación es de bajo voltaje entre 2.0 a 3.6V, tiene un rango de medición ajustable entre $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$, y una alta resolución de hasta 13 bits, con una sensibilidad de 40mg/LSB en todos los rangos (Volt, 2016).

En la figura 2.10 se observa la ilustración del sensor ADXL345, los pines de conexión y los diferentes microcomponentes.

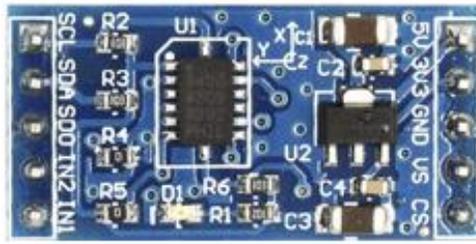


Figura 2.10. Sensor acelerómetro MMA7361

Fuente: (Llamas, 2018)

2.3.11. SENSOR BMG160

BMG160 es un sensor de giroscopio, básicamente un sensor de giroscopio triaxial digital de 16 bits que puede medir velocidades angulares. Es capaz de calcular velocidades angulares en tres dimensiones de habitación perpendiculares, los ejes x, y, z, y proporcionar las señales de salida correspondientes. Puede comunicarse con la placa Raspberry Pi mediante el protocolo de comunicación I2C. Este módulo en particular está diseñado para cumplir con los requisitos de aplicaciones de consumo, así como con fines industriales (Arduino, 2016).

El protocolo de comunicación sobre el que trabaja el sensor es I2C. I2C significa el circuito interintegrado. Es un protocolo de comunicación en el que la comunicación se realiza a través de líneas SDA datos en serie y SCL reloj en serie (Arduino, 2016).

En la figura 2.11 se observa la ilustración del sensor BMG160, así mismo las conexiones del sensor.

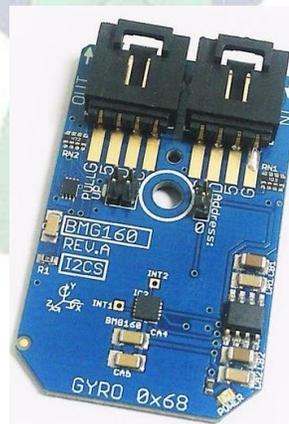


Figura 2.11. Sensor giroscopio BMG160

Fuente: (ARDUINO, 2016)

2.3.12. SENSOR ITG-3200

ITG-3200 es un sensor giroscopio de tres ejes de salida digital, incluye tres convertidores analógicos X, Y y Z de velocidades angulares en un solo circuito integrado a digital de 16 bits, una interfaz de dos hilos I2C. Las características adicionales incluyen un sensor de temperatura integrado y un oscilador interno de precisión 2%. El sensor puede ser alimentado con tensiones de entre 2.1v y 3.6V. en funcionamiento normal del sensor el consumo es solo 6.5mA, para larga duración de funcionamiento con baterías (InvenSense, 2011).

En la figura 2.12 se observa la ilustración del sensor.

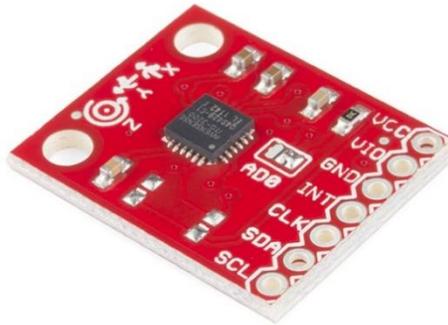


Figura 2.12. Sensor giroscopio ITG-3200

Fuente: (<https://learn.sparkfun.com/tutorials/itg-3200-hookup-guide/all>)

2.4. NODEMCU

NodeMCU es un nombre que recoge tanto un firmware Open Source y como a una placa de desarrollo basados en el ESP8266, que utiliza el lenguaje de programación Lua para crear un ambiente de desarrollo propicio para aplicaciones que requiera conectividad Wifi de manera rápida (Llamas, 2018).

El firmware NodeMCU podía grabarse en un ESP8266, tras lo cual podíamos programarlo con el lenguaje script Lua. La programación el Lua permitía la conexión y programación del ESP8266 de una forma mucho más sencilla que las herramientas oficiales proporcionadas por Espressif (Llamas, 2018).

El firmware NodeMCU fue creado poco después de aparecer el ESP8266, el 30 de diciembre de 2013. Unos meses después, en octubre de 2014 se publicó la primera versión del firmware NodeMCU en Github. Dos meses más tarde se publicaba la primera placa de desarrollo NodeMCU, denominada devkit v0.9, siendo también Open Hardware.

Con el paso del tiempo y la aparición de otras alternativas para programar ESP8266, como (especialmente) con C++ con el entorno del Arduino y otras como MicroPython, el interés en Lua ha disminuido considerablemente (Llamas, 2018).

2.5. PLACA DE DESARROLLO NODEMCU

NodeMCU es una placa de desarrollo totalmente abierta, a nivel de software y de hardware. Al igual que ocurre con Arduino, en NodeMCU todo está dispuesto para facilitar la programación de un microcontrolador o MCU³ (Hernández, 2018).

En la figura 2.13 se muestra la placa NodeMCU, pero, además añade las siguientes ventajas propias de placas de desarrollo:

- Puerto micro USB y conversor Serie-USB
- Programación sencilla a través del Micro-USB
- Alimentación a través del USB
- Terminales (pines) para facilitar la conexión
- LED y botón de reset integrados

NodeMCU es un nombre que recoge tanto un firmware Open Source y como a una placa de desarrollo basados en el ESP8266 Llamas (2018).



Figura 2.13. Placa de desarrollo NodeMCU

Fuente: (Llamas, 2018)

2.6. INTERNET DE LAS COSAS IOT

EL Internet de las Cosas es “la interconexión a través de Internet de dispositivos informáticos integrados en objetos cotidianos, lo que les permite enviar y recibir datos”. En otras palabras, IoT conecta tus dispositivos a Internet o a otros aparatos, para que puedan realizar nuevas funciones, como por ejemplo controlar elementos inteligentes de forma remota

³ Siglas del inglés Microcontroller Unit

y recibir alertas y actualizaciones de estado. Se refiere a los miles de millones de dispositivos físicos en todo el mundo que ahora están conectados, recolectando y compartiendo datos (Peña , 2019).

Gracias a las redes inalámbricas y el bajo costo de los nuevos procesadores, es posible que casi cualquier cosa, desde una aspiradora inteligente hasta un vehículo autónomo, forme parte de la IoT. Esto agrega un nivel de inteligencia digital a los dispositivos que les permite comunicar datos en tiempo real sin la participación de un ser humano, fusionando de alguna manera el mundo digital con el físico (Peña , 2019).

2.7. COMO FUNCIONA IOT

Según Valois (2018) donde indica que el Internet de las Cosas trata de objetos conectados entre sí por medio de la red. Estos intercambian información para facilitar o crear diversas acciones. Para que algo así pueda ocurrir hay un conjunto de tres factores que necesitan ser combinados para que una aplicación funcione dentro del concepto de Internet de las Cosas. Estos son: dispositivos, la red y un sistema de control.

- **Los Dispositivos:** Son todos aquellos que ya conocemos, como: refrigeradores, carros, lámparas, relojes, cafeteras, televisión y otros. En estos dispositivos es importante que sean equipados con los ítems correctos para proporcionar la comunicación con los demás elementos. Esos límites pueden ser chips, conexión con internet, sensores, antenas entre otros.
- **La Red:** Es el medio de comunicación y ya estamos acostumbrados a ella. Pues son tecnologías como Wi-Fi, Bluetooth y datos móviles (3G y 4G).
- **El sistema de control:** es necesario para que todos los datos capturados de los dispositivos a través de red sean procesados, entonces ellos son enviados para un sistema que controla cada aspecto y hace nuevas conexiones.

2.8. APLICACIONES DE IOT

Según del Valle Hernández (2017) el cual menciona los sectores donde se aplica Iot como son:

2.8.1. APLICACIONES DEL IOT PARA EL SECTOR INDUSTRIAL

Este sector cubre una amplia gama de servicios y productos. En muchos de ellos se puede utilizar el IoT de diferentes maneras.

Hostelería: En este sector ya se está utilizando dispositivos del IoT. Seguramente ya hayas visto como algunos restaurantes de comida rápida tienen un terminal que te avisa cuando está lista tu comida. Incluso los camareros son capaces de localizar la mesa a dónde va la comida. Si miramos desde el lado de la gestión, los almacenes de comida tienen mucho que decir. Previene la caducidad de alimentos, necesidades, pedidos automáticos y gestión de aparatos como cámaras frigoríficas, cocinas, hornos, etc.

Comercios de cara al público: Gracias a las aplicaciones del IoT, esto se está trasladando a los comercios de venta al público. Sobre todo, lo podemos encontrar en los centros comerciales. Preguntas como ¿cada cuánto tiempo viene un cliente? ¿a qué horas? ¿qué compra? ¿desde dónde viene? ¿cuánto tiempo pasa comprando? se pueden responder con sensores. Y la realidad es que se está haciendo.

Flotas de vehículos para logística: La logística es uno de los sectores donde más influencia tienen las aplicaciones del IoT. Controlar los paquetes, gestionar a los vehículos, evitar robos y gestión de movimientos son algunos aspectos a tener en cuenta dentro de esta área.

Gestión de almacenes de logística: Si antes hablaba de controlar los vehículos, imagínate como serán los almacenes donde están todas esas mercancías. Es indispensable poder controlar cada uno de ellos para que sean capaces de llegar a tiempo a su destino. Esto no es algo nuevo. Desde hace mucho tiempo (1996) se utilizan estándares como el EDI para el intercambio electrónico de datos. Este estándar está basado en código de barras. Las grandes empresas del transporte como DHL tienen departamentos de investigación en nuevas tecnologías.

Agricultura y ganadería: En la agricultura y ganadería también se están utilizando aplicaciones del IoT. Sobre todo, para hacer seguimiento de magnitudes como la temperatura, humedad, luminosidad y demás factores que pueden influir en la producción. Todo esto facilita a los agricultores el poder predecir y cuantificar cada cosecha antes de recogerla.

2.8.2. APLICACIONES DEL IOT PARA EL USO DOMESTICO

Poder crear nuestras propias aplicaciones del IoT para usar en nuestras casas.

- **Domótica:** Uno de los grandes retos de la tecnología es la domótica. Todavía no se ha conseguido despegar en este sentido debido a la dificultad en las infraestructuras

y la ausencia de estándares que permitan comunicar entre aparatos de diferentes marcas.

- **Jardinería:** El IoT nos puede aportar información muy valiosa en este sentido, parámetros que nos permitan identificar deficiencias en nuestras plantas, por ejemplo. Regadío automático e incluso automatizar la plantación.
- **Salud:** Las aplicaciones del IoT se utilizan para la telemedicina, control de pacientes en tiempo real, diagnósticos anticipados, etc.

2.8.3. APLICACIONES DEL IOT EN CIUDADES INTELIGENTES

Las ciudades inteligentes o smart cities, son los lugares donde más notaremos la introducción de aplicaciones del IoT. Se pretende resolver problemas que en la actualidad están afectando a todas las ciudades del mundo.

- **Gestión de suministros:** Poder controlar la gestión de suministros como agua, electricidad o gas es uno de los retos de esta tecnología.
- **Gestión ambiental:** Controlar la calidad del aire es relativamente sencillo gracias a Arduino. La cuestión es que de poco vale si no se ataja el problema de raíz. En este caso, de poco sirve la captación y análisis de esta información si no se crea una conciencia ciudadana. El agua también puede sufrir la contaminación. La empresa española Libelium nos ofrece el sistema Smart Water que permite el análisis y gestión del agua.
- **Gestión del tráfico:** Está muy vinculado a la contaminación. Una gestión eficiente del tráfico puede evitar niveles de contaminación altos. Aquí Google ya nos ofrece Maps, donde en todo momento nos informa de la situación del mismo. Son aplicaciones del IoT, aunque nosotros no lo sepamos. Usan los datos anónimos de los usuarios para enviar la posición y velocidad a través del móvil y poder mostrar la información de las carreteras.

2.9. PLATAFORMA IOT

Una plataforma IoT es la base para que dispositivos estén interconectados y se genere un ecosistema propio. Dicho de otra forma y según Link-labs, una plataforma web integrada al Internet of Things (IoT) es el software que conecta hardware, puntos de acceso y redes de

datos a lo que generalmente suele ser la aplicación de la que disfruta el usuario. (Cárdenas, 2016).

En pocas palabras, una plataforma IoT es el software capaz de conectar todo en un sistema IoT. Según Holloway (2018) un ecosistema IoT está compuesto por 4 elementos claves IoT:

Hardware: como sensores o dispositivos, que recolectan información del entorno (detectar cuándo hace falta cambiar la pieza de una máquina) o desempeñan acciones en el mismo (cambiar la temperatura ambiente para mantenerla dentro de márgenes predefinidos).

Conectividad: el hardware necesita una forma de transmitir toda esa información a la nube, así como también un camino para recibir órdenes desde la nube. En algunos sistemas IoT puede existir también un router o gateway intermedio entre los dispositivos y la nube: ahí entra a operar, por ejemplo, el Edge Computing.

Software: por lo general el software de un sistema IoT completo está en la nube y es responsable de analizar la información que está enviando el hardware con el objetivo de tomar decisiones.

Interfaz de usuario: a fin de que todo lo anterior tenga una utilidad real debe existir un método para que los usuarios interactúen con el sistema IoT, sean capaces de monitorearlo y tomar decisiones a partir de la información que ahí consiguen. Este es el punto donde se genera potencialmente más valor al negocio.

2.10. FIREBASE

Es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo (López, 2020).

Aunque fue creada en 2011 pasó a ser parte de Google en 2014, comenzando como una base de datos en tiempo real. Sin embargo, se añadieron más y más funciones que, en parte, permitieron agrupar los SDK de productos de Google con distintos fines, facilitando su uso (López, 2020).

Según Ruiz (2017) mencionó los servicios más destacados que ofrece Firebase para trabajar de una manera más sencilla en las aplicaciones las cuales son las siguientes.

Base de datos Real time: Con la base en tiempo real de Firebase se guardan todos los datos que requiera la aplicación. Los datos se almacenan en formato JSON y se pueden agregar reglas para permitir requests con token o solo desde una URL por ejemplo.

Autenticación: Es un servicio que simplifica el inicio de sesión y la gestión de la misma en la aplicación mediante el email y las redes sociales.

Almacenamiento: Este servicio es muy bueno para aplicaciones que requieran guardar archivos del usuario como guardar imágenes entre otros.

Hosting: Se usa para publicar página web, subir una aplicación y esta automáticamente contará con SSL y HTTP2.

Laboratorio de test para Android: Este servicio ofrece la posibilidad de probar los test de nuestra aplicación en los entornos que configuremos antes de la publicación.

Informes sobre fallos: Con esta utilidad permite detectar los errores que aparezcan en las aplicaciones de iOS y Android, así como poder erradicarlos a tiempo.

Notificaciones: Este servicio permite gestionar el envío de notificaciones a usuarios con la diferencia de que estas podrán ser programadas acorde a diferentes parámetros.

Configuración remota: Este añadido permite básicamente modificar el comportamiento y la apariencia de la aplicación sin los usuarios tengan que tocar nada.

Enlaces dinámicos: Con los enlaces dinámicos se asegura que a pesar de las redirecciones que supone el proceso de instalar una aplicación desde un anuncio, se llega a la vista deseada, desde otras aplicaciones.

Admob: Este servicio permite colocar publicidad en distintos lados de la aplicación.

En la figura 2.14 se muestra los servicios que ofrece Firebase.

2.11. JSON

JSON, cuyo nombre corresponde a las siglas JavaScript Object Notation o Notación de Objetos de JavaScript, es un formato ligero de intercambio de datos, que resulta sencillo de leer y escribir para programadores y simple de interpretar y generar para las máquinas.

JSON es un formato de texto completamente independiente de lenguaje, pero utiliza convenciones que son ampliamente conocidos, entre ellos: C, C++, C#, Java, JavaScript, Perl Python, entre otros.

Dichas propiedades hacen de JSON un formato de intercambio de datos ideal para usar con API REST o AJAX. A menudo se usa en lugar de XML, debido a su estructura ligera y

compacta. Muchos lenguajes de programación proporcionan métodos para analizar una cadena de texto con este formato en un objeto nativo y viceversa.



Figura 2.14. Servicios de Firebase

Fuente: (Zamora, 2016)

Según la descripción de Stack Overflow, JSON “define seis tipos de valores: nulo, números, cadenas, booleanos, matrices y objetos”.

Pese a su nombre, no es necesariamente parte de JavaScript, de hecho, es un estándar basado en texto plano para el intercambio de datos, por lo que se usa en muchos sistemas que requieren mostrar o enviar información para ser interpretada por otros sistemas.

Una de las características de JSON, al ser un formato que es independiente de cualquier lenguaje de programación, es que los servicios que comparten información por este método no necesitan hablar el mismo idioma, es decir, el emisor puede ser Java y el receptor Python, pues cada uno tiene su propia librería para codificar y decodificar cadenas en este formato (Barrera, 2016).

2.12. NoSQL

NoSQL es una base de datos no relacional que almacena y accede a datos utilizando valores-clave. En lugar de almacenar datos en filas y columnas como una base de datos tradicional, un DBMS NoSQL almacena cada elemento individualmente con una clave única.

Además, una base de datos NoSQL no requiere un esquema estructurado que defina cada tabla y las columnas relacionadas. Esto proporciona un enfoque mucho más flexible y escalable para almacenar datos que una base de datos relacional.

NoSQL proporciona un enfoque no estructurado o "semiestructurado" que es ideal para capturar y almacenar contenido generado por el usuario. Esto puede incluir texto, imágenes, archivos de audio, videos, transmisiones de clics, tweets u otros datos. De hecho, puede agregar miles o cientos de miles de registros nuevos a una base de datos NoSQL con una disminución mínima del rendimiento. Son especialmente comunes en aplicaciones de computación en la nube y se han convertido en la solución de almacenamiento más popular para big data (Christensson, 2013).

2.13. ANDROID STUDIO

Ferreño (2020) explica que Android Studio es un entorno de desarrollo, un software, que cuenta con herramientas y servicios para que los desarrolladores puedan crear nuevas aplicaciones para Android. Así también menciona que muchos de los sistemas operativos actuales cuentan con este tipo de entornos de desarrollo, algo que ocurre también en el sistema operativo de Google.

Por otro lado, también menciona que este entorno de desarrollo como se muestra en la imagen XX, se encuentran con las herramientas que son necesarias para crear aplicaciones. Esto incluye desde el código, al diseño de la interfaz de usuario de la aplicación. Va a poder guiar a dicho creador en todo el proceso, para poder obtener de esta manera una aplicación, que se podrá publicar posteriormente en la Play Store, por ejemplo.

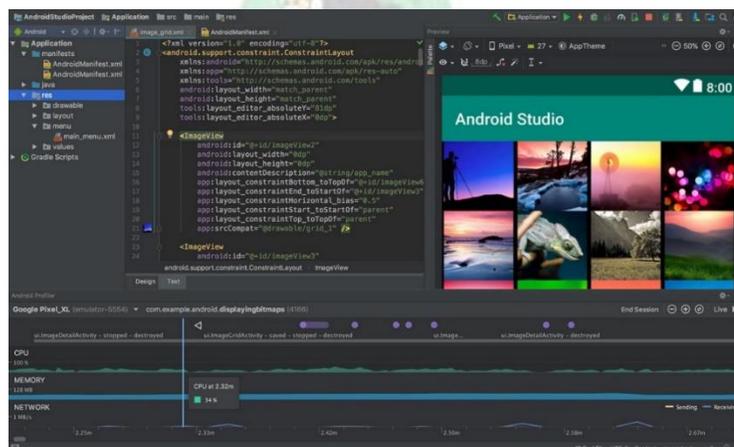


Figura 2.15. Entorno de desarrollo Android Studio

Fuente: (Ferreño, 2020)

2.14. ADULTO MAYOR

El concepto de adulto mayor presenta un uso relativamente reciente, ya que ha aparecido como alternativa a los clásicos persona de la tercera edad y anciano. En tanto, un adulto mayor es aquel individuo que se encuentra en la última etapa de la vida, la que sigue tras la adultez y que antecede al fallecimiento de la persona. Porque es precisamente durante esta fase que el cuerpo y las facultades cognitivas de las personas se van deteriorando (Ucha, 2013).

2.15. CAÍDA

Según Organización mundial de la salud (2018) las caídas se definen como acontecimientos involuntarios que hacen perder el equilibrio y dar con el cuerpo en tierra u otra superficie firme que lo detenga.

2.15.1. CLASIFICACIÓN DE LAS CAÍDAS

Según Cruz & Pérez (2014) donde indica que de acuerdo con criterios de causalidad y tiempo de permanencia en el piso, las caídas se pueden clasificar en tres.

- **Caída accidental:** Es aquella que se produce por una causa ajena al adulto(a) mayor, con origen en un entorno potencialmente peligroso, por ejemplo, un tropiezo con un objeto o barrera arquitectónica.
- **Caída de repetición "no justificada":** Es aquella donde se hace patente la persistencia de factores predisponentes como polipatología o polifarmacia. Un ejemplo lo constituyen la enfermedad de Parkinson y la sobredosificación con benzodiazepinas.
- **Caída prolongada:** Es aquella en la que el adulto(a) mayor permanece en el suelo por más de 15 a 20 minutos con incapacidad de levantarse sin ayuda. Las caídas con permanencia prolongada en el piso son indicativas de un mal pronóstico para la vida y la función.

2.15.2. ANÁLISIS DE UNA CAÍDA

La aceleración y orientación del cuerpo de una persona que se encuentra de pie y se cae, pasa por distintos estados desde el momento en que pierde el equilibrio hasta que se encuentra en el piso. En la figura 2.16 se pueden observar los cuatros etapas por las que pasa una persona cuando experimenta una caída. Inicialmente una persona se encuentra en Estado de reposo inicial (A), dicha posición se caracteriza porque la persona se encuentra en un ángulo inicial

de la caída. Luego cuando la persona pierde el equilibrio, entra en una caída libre (B), hasta impactar con el suelo (C). Una vez que se produjo el impacto la persona se encuentra un tiempo determinado en reposo, hasta que se pone de pie nuevamente. Durante ese intervalo de tiempo la persona adquiere un determinado ángulo final (D), con respecto al ángulo inicial (Carnuccio, y otros, 2017).

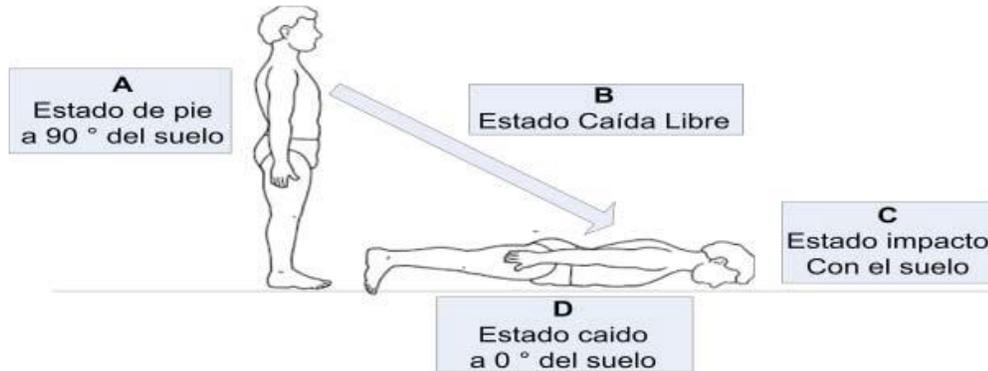


Figura 2.16. Etapas de la caída

Fuente: (Carnuccio, y otros, 2017)

2.16. METODOLOGÍA V

El Método-V es una representación gráfica del ciclo de vida del desarrollo del sistema. Resume los pasos principales que hay que tomar en conjunción con las correspondientes entregas de los sistemas de validación.

La parte izquierda de la V representa la corriente donde se definen las especificaciones del sistema. La parte derecha de la V representa la corriente donde se comprueba el sistema (contra las especificaciones definidas en la parte izquierda). La parte de abajo, donde se encuentran ambas partes, representa la corriente de desarrollo (Mex, 2018).

La corriente de especificación consiste principalmente de:

- Especificaciones de requerimiento de usuario
- Especificaciones funcionales
- Especificaciones de diseño

La corriente de pruebas, por su parte, suele consistir de:

- Calificación de instalación
- Calificación operacional
- Calificación de rendimiento

Los proyectos lleguen finalmente a ser exitosos desde los puntos de vista de objetivos de negocio, costos, funcionalidad, sencillez y capacidad de soporte. La corriente de desarrollo puede consistir (depende del tipo de sistema y del alcance del desarrollo) en personalización, configuración o codificación (Mex, 2018).

En general las metodologías llevan a cabo una serie de procesos comunes que son buenas prácticas para lograr los objetivos antes mencionados independientemente de cómo hayan sido diseñadas. Las fases que agrupan estos procesos son las siguientes:

- **Fase 1. Especificaciones:** Se debe definir y documentar los diferentes requisitos del sistema a desarrollar, identificar los valores numéricos más concretos posibles. Especificación estructurada utilizando diferentes técnicas de diagramas para modelar el sistema nuevo.
- **Fase 2. Funcional:** También llamado de alto nivel. Su objetivo es obtener un diseño y visión general del sistema. Establecer un conjunto de módulos, desplegando las necesidades obtenidas en la fase de especificaciones, facilitando a la tarea de codificación y los modelos lógicos de los datos físicos.
- **Fase 3. Diseño:** Consiste en detallar cada módulo de la fase anterior. Cada módulo como resultado de la fase anterior es producido en la herramienta o lenguaje apropiado. Define los componentes de hardware y software del sistema final a cuyo conjunto se denomina arquitectura del sistema.
- **Fase 4. Codificación:** En esta fase se materializa el diseño en detalle, se implementan, desarrollan los elementos unitarios o módulos del programa.
- **Fase 5. Test de Diseño:** Se verifica cada módulo desarrollado de forma unitaria, comprobando su funcionamiento adecuado, generando documentos de las pruebas. Es la verificación del correcto funcionamiento del hardware y software, con los módulos producidos por las herramientas o lenguajes escogidos.
- **Fase 6. Test Funcional:** Es la fase de prueba de la funcionalidad en el que se integran los distintos módulos que forman el sistema. Como en el caso anterior, ha de generarse un documento de pruebas. Por una parte, se debe comprobar en todo el sistema tenga el funcionamiento correcto, y por otra, en caso de tratarse con un sistema tolerante a fallos, se debe verificar que, ante la presencia de alguno, persista

el funcionamiento correcto. Se comprueba el cumplimiento de los requisitos establecidos.

- **Fase 7.** Test de Especificaciones: Se realizan las últimas pruebas, pero sobre un escenario real, en su ubicación final, anotando una vez más las pruebas realizadas y los resultados obtenidos. Garantizar el mantenimiento del sistema, corrección de errores detectados en la fase, probando la aplicación del sistema en nuevos entornos.

La metodología en V se trata de un proceso ideal, por su robustez, para proyectos pequeños, con equipos de una a cinco personas.

También es ideal, por su claridad, para toda esa gente que nunca ha programado siguiendo una metodología.

En la figura 2.17 se presenta el Modelo en V, o Modelo de Cuatro Niveles, del ciclo de vida de un proyecto de desarrollo de software. El modelo representa, en forma de V, las relaciones temporales entre las distintas fases del ciclo de desarrollo de un proyecto (Mex, 2018).

En los niveles lógicos del 1 al 4, para cada fase del desarrollo, existe una fase correspondiente o paralela de verificación o validación. Esta estructura obedece al principio de que para cada fase del desarrollo debe existir un resultado verificable (Lariza & Perez, 2012).

En la misma estructura se advierte también que la proximidad entre una fase del desarrollo y su fase de verificación correspondiente va decreciendo a medida que aumenta el nivel dentro de la V. La longitud de esta separación intenta ser proporcional a la distancia en el tiempo entre una fase y su homóloga de verificación (Lariza & Perez, 2012).

- **El nivel 1.** Está orientado al cliente, el inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo. Se compone del análisis de requisitos y especificaciones, se traduce en un documento de requisitos y especificaciones.
- **El nivel 2.** Se dedica a las características funcionales del sistema propuesto, puede considerarse el sistema como una caja negra, y caracterizarla únicamente con aquellas funciones que son directa o indirectamente visibles por el usuario final, se traduce en un documento de análisis funcional.
- **El nivel 3.** define los componentes hardware y software del sistema final, a cuyo conjunto se denomina arquitectura del sistema.

- **El nivel 4.** Es la fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa.

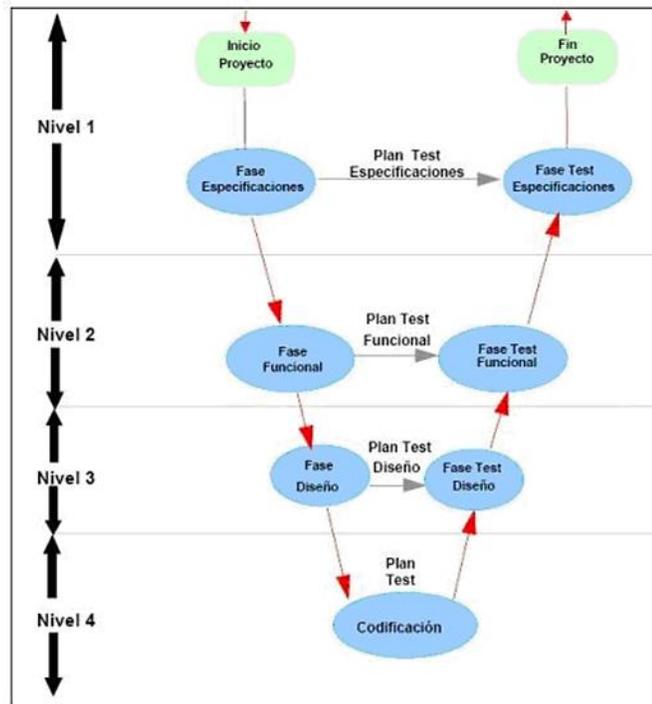


Figura 2.17. Ciclo de vida de un proyecto de desarrollo de software

Fuente: (Mex, 2018)

2.17. FRITZING

Fritzing es un programa de diseño gratuito que se puede utilizar para diseñar circuitos en placas de prueba, crear esquemas e incluso desarrollar PCB. A diferencia de muchas otras suites de diseño de circuitos, Fritzing está pulido y utiliza en gran medida imágenes vectorizadas, produciendo circuitos de aspecto hermoso.

Si bien Fritzing incluye muchos componentes populares, como la gama de placas y módulos Arduino, está diseñado para uso aficionado y de fabricante en lugar de uso profesional, en la figura 2.18 se muestra el logo de Fritzing (Mitchell, 2018).



Figura 2.18. Logo de Fritzing

Fuente: (Fritzing, 2018)

3. DISEÑO METODOLÓGICO

3.1. INTRODUCCIÓN

En el presente capítulo se realizará el desarrollo de hardware y software para la implementación del prototipo para la detección de caídas en adultos mayores, haciendo el uso de la metodología descrita en la sección 2.16, para así poder conseguir los objetivos propuestos en el capítulo I y lograr un mejor funcionamiento del prototipo.

3.1.1. PROCEDIMIENTOS DEL MODELO EN V

El modelo en V, consta de pasos y fases para realizar el diseño del prototipo detector de caídas en adultos mayores, para tal caso se formula las fases del modelo en V que se describe a continuación.

Fase 1. Especificaciones: En esta fase se define y documenta los requisitos del prototipo detector de caídas, identificando las variables que se controlaran para así poder obtener diagramas para el modelado.

Fase 2. Funcional: En esta fase se tiene como objetivo tener un diseño, un esquema general del prototipo.

Fase 3. Diseño: En esta fase consiste en detallar y describir los componentes tanto de hardware como de software necesarios para desarrollar el prototipo.

Fase 4. Codificación: En esta fase se materializa el diseño del prototipo detector de caídas, donde se indica cada uno de los elementos que determina el proceso componiendo el funcionamiento de software y hardware.

Fase 5. Test de diseño: En esta fase se verificará cada uno de los módulos desarrollados, comprobando el correcto funcionamiento adecuado del hardware y software del prototipo.

Fase 6. Test funcional: En esta fase se realizarán pruebas de funcionalidad en las que se integrarán los diversos módulos que compone el prototipo, tanto el hardware y software.

Fase 7. Teste de especificaciones: En esta última fase se realiza pruebas de funcionamiento sobre un escenario real, la ubicación apropiada del prototipo, registrando las pruebas realizadas y resultados obtenidos.

3.2. FASE 1. ESPECIFICACIONES

La especificación para el prototipo detector de caídas depende del tipo de plataforma, sensores y los módulos de comunicación que serán usadas para desarrollar el prototipo, básicamente consiste en dos elementos principales, es decir la implementación tanto hardware como el software el cual se detalla a continuación.

Hardware: Recolectar información de un componente externo, obteniendo datos del acelerómetro, el giroscopio, los cuales se procesarán y con estos datos determinar un evento de caída y posteriormente generar una alarma sonora y una alarma inalámbrica para lo cual necesariamente tiene que estar conectado al internet para el envío de la notificación de alarma.

Software: Aplicación móvil estará conectado a internet, la función que tiene es de recibir las notificaciones de alerta que enviara el hardware tras una detección de una situación de caída que haya sufrido la persona.

3.2.1. DIAGRAMA PARA EL MODELADO DE REQUISITOS

La fase uno del modelo en V, adoptará el siguiente diagrama que forma parte del UML, el cual es una herramienta para el diseño de sistemas los cuales representan como deberá interactuar el sistema con el usuario o con otros sistemas en respuesta a un evento para conseguir un objetivo.

3.2.2. DIAGRAMA DE CASOS DE USO.

A continuación, se observará el diagrama de casos de uso donde el usuario y el sistema podrá realizar las respectivas actividades de cada actor que se tiene el cual se muestra a en la figura 3.1.

Como parte de la documentación, posteriormente se elaboró para cada proceso en el diagrama el respectivo escenario de casos de uso el cual permite comprender mejor el comportamiento del prototipo detector de caídas, se describirán a los procesos que realizan los actores como son el usuario y sistema.

El usuario principal al que está destinada este dispositivo que es el adulto mayor puesto que el será que llevará puesto el prototipo consigo.

Como usuario secundario llegarían a ser los hijos, cuidador, apoderados quienes serán las personas que podrán recibir las notificaciones de alarma en la aplicación cuando sufra una caída el adulto mayo.

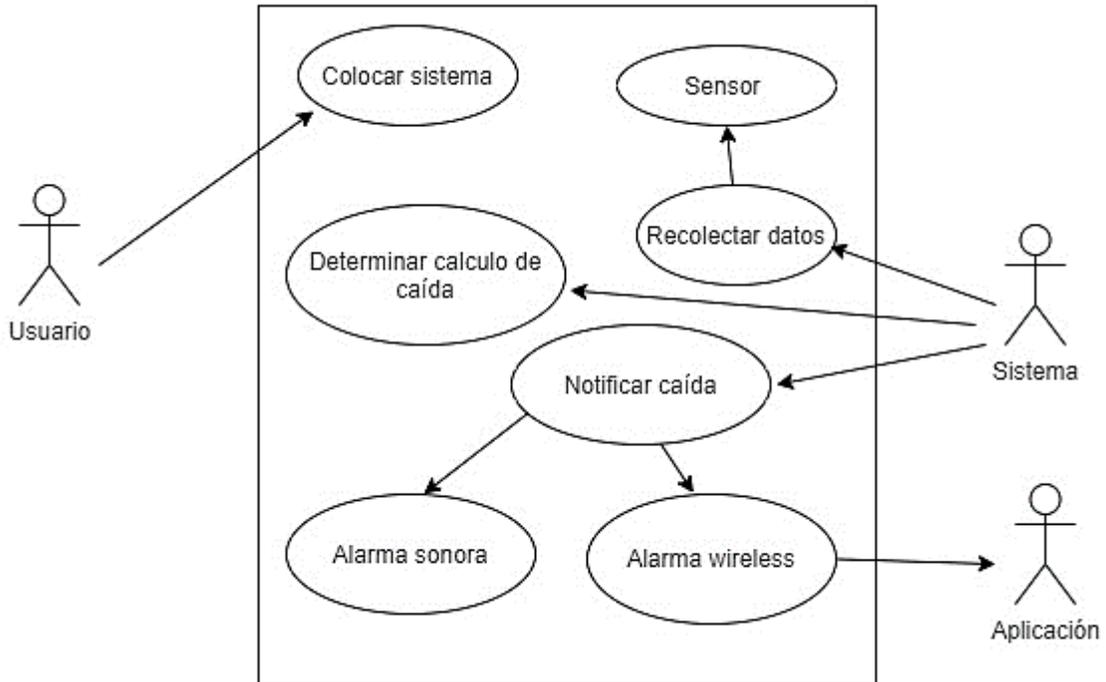


Figura 3.1. Diagrama casos de uso del prototipo detector de caídas

Fuente: Elaboración propia

En la tabla 3.1 se muestra el primer escenario que involucra al usuario, donde se detalla la descripción del caso de uso y cual las funciones que deben tener para cumplir.

Tabla 3.1. Descripción casos de uso: colocar sistema

Caso de uso	Colocar sistema
Actores	Usuario
Descripción	Colocar el dispositivo a la persona adulta mayor a monitorear con el cual se detectará las caídas y se activaran las alarman ante un evento de ese tipo.
Evento desencadenador	El usuario pone en funcionamiento el sistema.
función	Recolección de información.

Fuente: Elaboración propia

A continuación, se muestran los escenarios de casos de uso que involucran al sistema, en la tabla 3.2 se describe el escenario de casos de uso recolectar datos para luego posteriormente hacer un cálculo de determinación de una caída, para luego hacer otro tipo de proceso que siguen.

Tabla 3.2. Descripción de casos de uso: recolectar datos

Caso de uso	Recolectar datos
Actores	Sistema
Descripción	Realiza el censado de coordenadas en los ejes x, y, z proporcionadas por parte del sensor.
Evento desencadenador	Puesto en funcionamiento el sistema.
función	Recolección de información.

Fuente: Elaboración propia

En la tabla 3.3 se describe el tercer caso de uso, el cual describe el cálculo de determinación de una caída luego de haber obtenido los datos del sensor.

Tabla 3.3. Descripción de casos de uso: determinar cálculo de caída

Caso de uso	Determinar cálculo de caída
Actores	Sistema
Descripción	Realiza cálculos de la magnitud de los tres ejes x, y, z es decir la aceleración y velocidad angular y con esto determinar si se produjo un evento de caída.
Evento desencadenador	La persona sufre cambios brusco en una de las actividad diarias.
función	Detectar un evento de caída en una persona.

Fuente: Elaboración propia

En la tabla 3.4 se describe el cuarto caso de uso, el cual describe las notificaciones que se producen el sistema al momento de detectar una caída.

Tabla 3.4. Descripción de casos de uso: notificar caída

Caso de uso	Notificar caída
Actores	Sistema y aplicación
Descripción	Después de realizar los cálculos correspondientes y determinar un evento de caída, se genera una alarma sonora, así también hace el envío de las notificaciones de alerta a la aplicación indicando que el usuario sufrió una caída y el cual necesita ser auxiliado inmediatamente.

Evento desencadenador	La persona sufrió una caída brusca.
función	Alarma sonora y notificar la detección de una caída a la aplicación.

Fuente: Elaboración propia

3.3.3 DIAGRAMA DE ACTIVIDAD

Se utilizan los diagramas de actividades del UML para modelar la interacción de los procesos del sistema.

En el diagrama que se observa en la figura 3.2 se detallará las distintas actividades que se realiza para una mejor manera de funcionamiento del prototipo detector de caídas, las actividades que tiene tanto el usuario como el sistema y la aplicación.

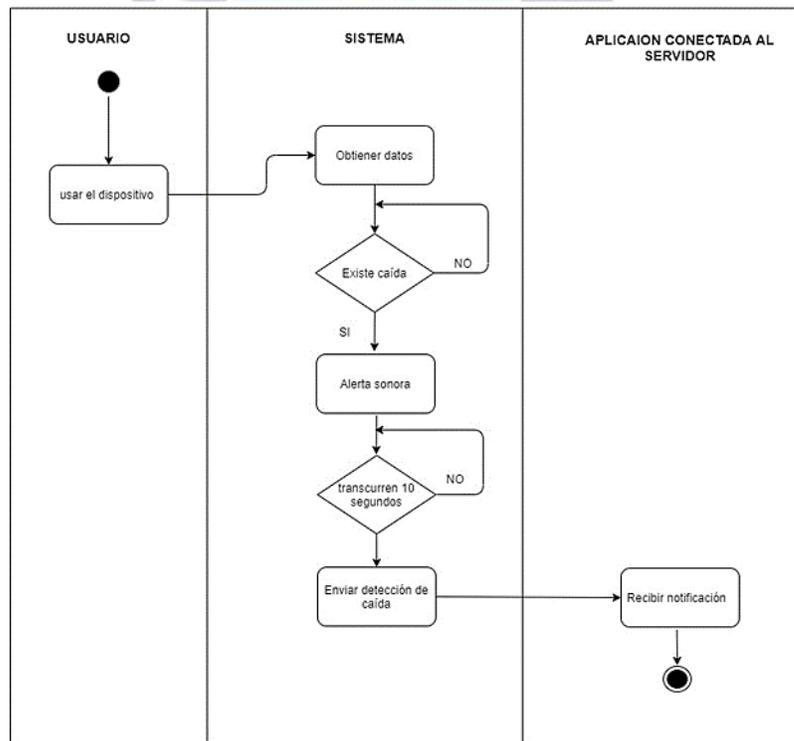


Figura 3.2. Diagrama de actividades del prototipo detector de caídas

Fuente: Elaboración propia

3.3. FASE 2. FUNCIONAL

También llamada diseño de alto nivel, su objetivo es obtener una visión general del sistema, en la figura 3.3 se describirá la funcionalidad del sistema para llegar al objetivo planteado en el capítulo 1.

En el prototipo detector de caídas contempla como partes principales la cual es una placa de desarrollo NODEMCU esp8266 para conectarse a una red wifi y enviar datos, un sensor inercial mpu6050 con el cual se obtiene datos tanto del acelerómetro y giroscopio con los cuales se emplea para la determinación de la existencia de una posible caída.

También se cuenta como partes secundarias, se tiene un zumbador el cual funcionara como una alarma sonora, también contara con un pulsador el cual servirá para cancelar falsas alarmas y en caso de que existieran, y por ultimo con una aplicación en donde se enviara las notificaciones de alarma correspondientes ante un evento de caída existente.

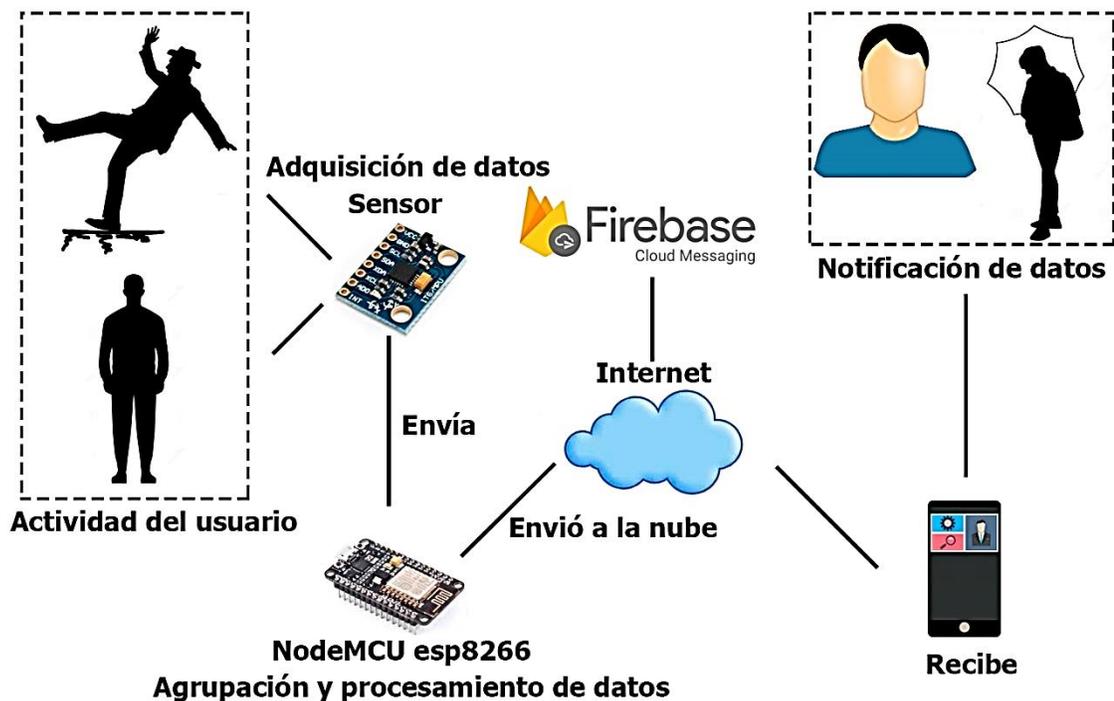


Figura 3.3. Descripción general del sistema detector de caídas

Fuente: Elaboración propia

3.4. FASE 3. DISEÑO

A continuación, en esta tercera fase se identifica, detallarán los diferentes componentes de software y hardware los cuales serán empleados para el desarrollo del dispositivo.

3.4.1. IDENTIFICACIÓN DE COMPONENTES DE HARDWARE

Módulo MPU-6050: Unidad inercial electromecánico de bajo consumo energético, cuenta con 6 grados de libertad (DOF) con las cuales se logra mediciones con el acelerómetro

y giroscopio en los tres ejes, este módulo de sensores permite medir las aceleraciones lineales y angulares, respectivamente a diferentes rangos de sensibilidad. Las especificaciones del MPU-6050 se encuentran en la tabla 3.5.

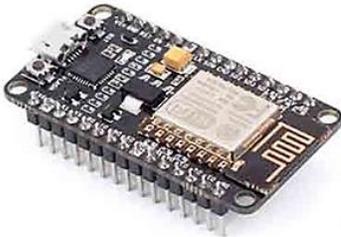
Tabla 3.5. Especificación del acelerómetro y giroscopio

Grafico	Características
	<ul style="list-style-type: none"> ✓ Salida digital de 6 ejes. ✓ Giroscopio con sensibilidad de ± 250, ± 500, ± 1000, y ± 2000dps ✓ Acelerómetro con sensibilidad de $\pm 2g$, $\pm 4g$, $\pm 8g$ y $\pm 16g$ ✓ Sensor de temperatura digital ✓ Voltaje de alimentación: 2.37 a 3.46V ✓ Voltaje lógico: $1.8V \pm 5\%$ o VDD ✓ 10000g tolerancia de aceleración máxima

Fuente: Elaboración propia

Módulo NodeMCU: Es una placa de desarrollo de código abierta, a nivel de software y de hardware, para el desarrollo todo está dispuesto para facilitar la programación de un microcontrolador, tiene acceso fácil a los pines entre otras ventajas lo que diferencia de los demás es que la placa está basada en el ESP8266 el cual permite conectar a una red wifi lo cual facilita el trabajo y de no comprar otro componente adicional, en la en la tabla 3.6 se encuentra las especificaciones del NodeMCU.

Tabla 3.6. Especificación del módulo NodeMCU

Grafico	Características
	<ul style="list-style-type: none"> ✓ Voltaje de entrada (USB): 5V ✓ Voltaje de salida en los pines: 3.3V ✓ Voltaje de referencia en el ADC: 3.3V ✓ Corriente nominal por pin: 12mA ✓ Frecuencia de procesador: 80MHz (160MHz max.) ✓ 4MB Flash

	<ul style="list-style-type: none"> ✓ Consumo de corriente en stand-by @80MHz: 80mA
--	---

Fuente: Elaboración propia

Módulo Buzzer / Zumbador: Dispositivo sonoro para producir la señal de alerta, emite un sonido a pulso de voltaje alto como indicador de alarma ante la caída de la persona, en la tabla 3.7 de detalla las especificaciones.

Tabla 3.7. Especificación del Zumbador

Grafico	Características
	<ul style="list-style-type: none"> ✓ Voltaje de operación 3.3 V a 5.5 V ✓ Oscilador Interno ✓ Dimensiones 12 mm diámetro – 9 mm altura ✓ Uso ON/OFF ✓ Pines VCC,GND

Fuente: Elaboración propia

Módulo Pulsador: Un botón o pulsador es un dispositivo utilizado para realizar cierta función, estos se activan al momento al ser pulsados por el dedo, el cual se utilizará para cancelar las falsas alarmas en caso de que existan estos, en la tabla 3.8 se detalla las especificaciones del pulsador.

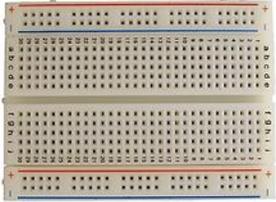
Tabla 3.8. Especificación del pulsador

Grafico	Características
	<ul style="list-style-type: none"> ✓ polarización alta la resistencia a + (5V) ✓ polarización baja la resistencia a masa – (0V) ✓ Tamaño: 6 x 6 x 5mm ✓ Temperatura de operación: -30 a + 70°C ✓ Voltaje: 250VAC / 12VDC ✓ Corriente: 50mA ✓ Resistencia de contacto: <0.03ohm ✓ Resistencia de aislamiento: >100Mohm

Fuente: Elaboración propia

Protoboard: Es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, el cual sirve para probar circuitos, armar el prototipo, entre otros, la tabla 3.9 de detalla el Protoboard.

Tabla 3.9. Especificación del Protoboard

Grafico	Características
	<ul style="list-style-type: none"> ✓ Vida útil de más 10.000 inserciones ✓ Soporte en plástico ABS ✓ 100 mils de separación estándar. ✓ Dimensiones: (16.5 x 5.5 x 0.8 cm) ✓ Protoboard de 830 puntos

Fuente: Elaboración propia

Cables: Nos servirá para hacer las conexiones de nuestros componentes electrónicos en el Protoboard sin la necesidad de hacer las soldaduras, en la tabla 3.10 se detallan los cables que se usaran.

Tabla 3.10. Especificación del cable

Grafico	Características
	<ul style="list-style-type: none"> ✓ Fácil de conectar y desconectar ✓ Aptos para pruebas ✓ diámetro: 22-26AWG ✓ material de aislamiento: PVC ✓ material conductor: cobre ✓ tipo: cable de conexión, macho a hembra

Fuente: Elaboración propia

3.4.2. IDENTIFICACIÓN DE COMPONENTES DE SOFTWARE

IDE Arduino: El lenguaje de programación de Arduino está basado en C++, es software de código abierto el cual se puede utilizar con cualquier placa Arduino, es multiplataforma para Windows, macOS, Linux, también se puede descargar desde la página oficial de Arduino, y además de las diferentes librerías que se usara en los módulos.

IDE Android Studio: Es el entorno de desarrollado integrado para el desarrollo de aplicaciones Android, el cual está disponible para plataformas Windows, macOS, y Linux,

con el cual se desarrolla una diversidad de aplicaciones, esta se lo puede descargar desde la página oficial de Android Studio.

Firestore Cloud Messaging: Si se quiere enviar o recibir un mensaje o notificación a la aplicación se debe usar el servicio Firestore Cloud Messaging (FCM) el cual es una plataforma en la nube perteneciente a google, es una mensajería multiplataforma segura y sin costos. Cuando una aplicación ya está configura para recibir notificaciones esta genera un código único que se llamara token, gracias a este token que funciona como un identificador el FCM puede enviar un mensaje a un usuario en específico o también se puede enviar mensajes a un grupo o a todo los que tienen la aplicación instalada.

3.4.3. DISEÑO DEL CIRCUITO ELECTRÓNICO

El circuito electrónico es la unión de todos los componentes, en el diseño del prototipo detector de caídas, se integran los componentes como el NodeMCU, de Zumbador, pulsador sobre todo lo más importante el sensor inercial MPU6050 el cual censará los datos de la aceleración y la velocidad angular y los enviará al microcontrolador el cual será procesado y determinará un evento de caída si existiera.

Todos estos elementos estarán conectados para un buen funcionamiento del sistema que en este caso es la detección de una caída.

En la figura 3.4 se muestra la conexión de los componentes ya mencionados anteriormente para lo cual se hará un diseño de las conexiones con la ayuda del software Fritzing.

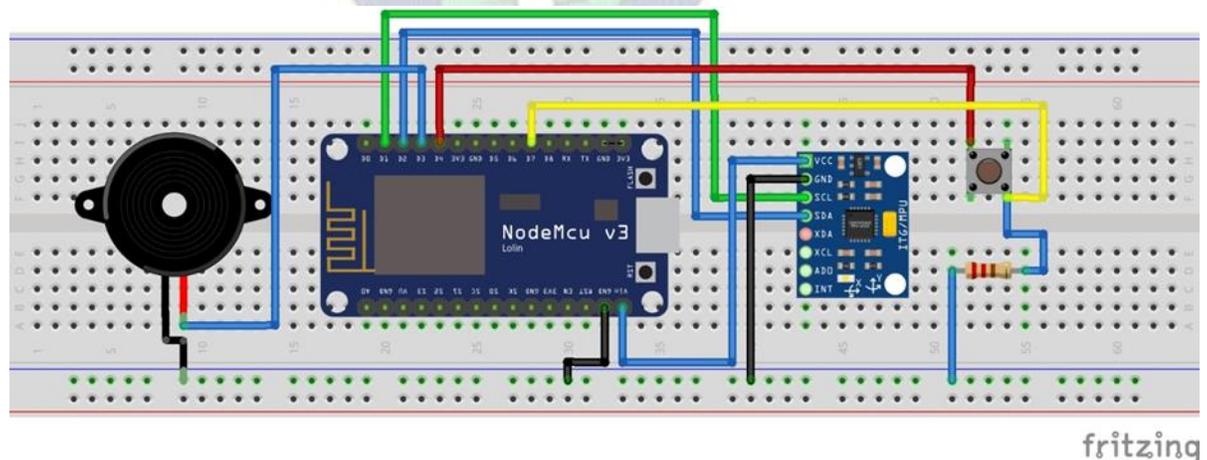


Figura 3.4. Conexión de los componentes

Fuente: Elaboración propia

En la Figura 3.5 se observa la representación del diagrama lógico del circuito electrónico MPU605, NodeMCU, zumbador, pulsador los cuales son los componentes y módulos interconectados que se refleja en el diagrama esquemático.

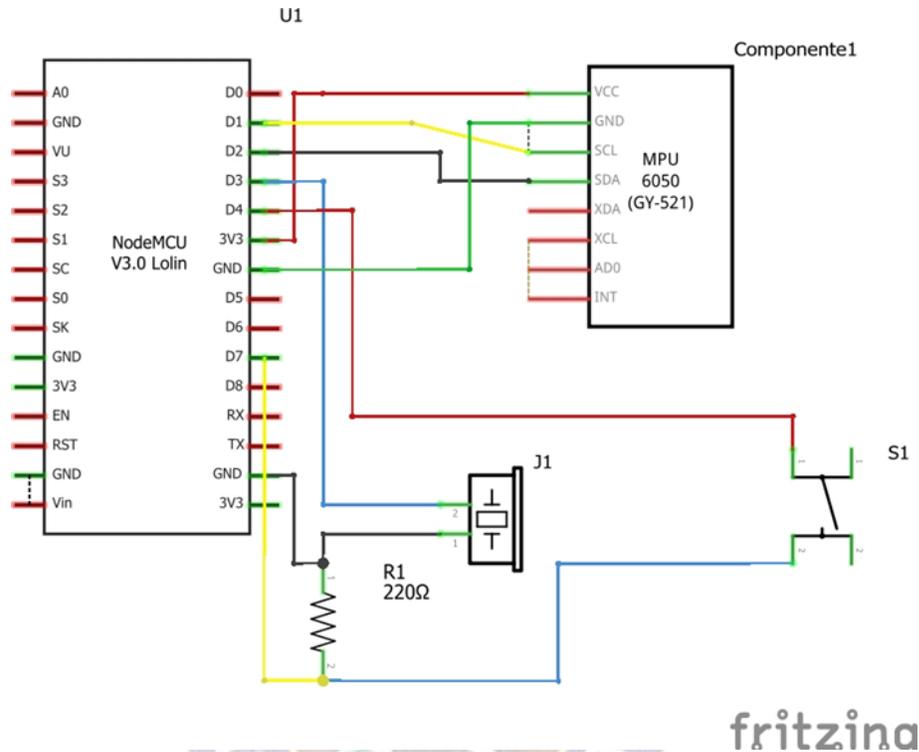


Figura 3.5. Diagrama lógico de los componentes

Fuente: Elaboración propia

3.5. FASE 4. CODIFICACIÓN

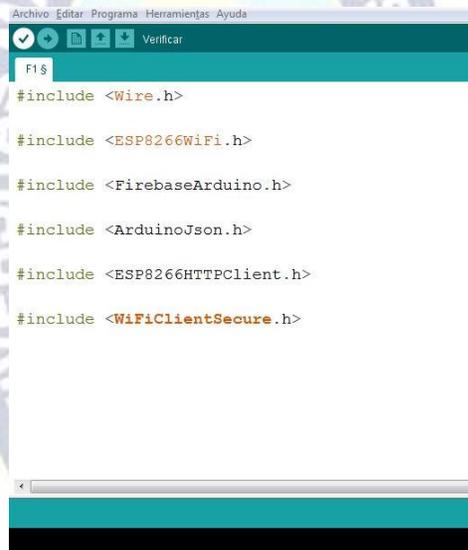
En esta fase describiremos el desarrollo del programa, se llevará acabo la programación del sistema, usando los componentes hardware y software mencionados anteriormente, y así de esta manera se plasmará las funcionalidades que se describió en diagramas de las fases anteriores, para lo cual se elaborara un plan de trabajo, una lista de especificaciones parecida al que se elaboró en la primera fase:

- ✓ Diseñar un sistema de recolección de datos como son las aceleraciones en los ejes ax, ay y az, así también velocidades angulares en los tres ejes gx, gy y gz, la orientación del sensor y analizar las mismas para determinar la existencia de una caída.
- ✓ Diseñar una aplicación móvil el cual recibirá la notificación enviada desde el NodeMCU al momento de producir una caída.

Antes de comenzar a programar se debe realizar las respectivas importaciones de todas las librerías necesarios para una buena comunicación, como ser:

- ✓ Inclusión de librerías para la comunicación serial.
- ✓ Inclusión de librerías para la comunicación I2C para el sensor inercial.
- ✓ Inclusión de librerías de comunicación con la placa NodeMCU.
- ✓ Inclusión de librerías para la conexión a wifi.
- ✓ Inclusión de librerías para la conexión a Firebase.
- ✓ Inclusión de librerías para la envío de peticiones post desde Arduino.

En la figura 3.6 corresponde al uso de las librerías necesarias que utiliza el sensor inercial, NodeMCU para la comunicación y la conexión a Firebase.



```
Archivo  Editar  Programa  Herramientas  Ayuda
Verificar
F1 $
#include <Wire.h>
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <ArduinoJson.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecure.h>
```

Figura 3.6. Librerías utilizadas para el NodeMCU y el sensor

Fuente: Elaboración propia

Las librerías importadas son fundamentales, y se debe realizar el uso de las librerías compatibles correspondientes para evitar posteriores problemas.

3.5.1. SISTEMA ADQUISICIÓN DE DATOS

Una de las especificaciones es la adquisición de datos, el cual se basa en el componente mencionado anteriormente el sensor MPU6050, el cual se encarga de medir la aceleración y la velocidad angular para los posteriores procesamientos de estos datos y con estos determinar una caída.

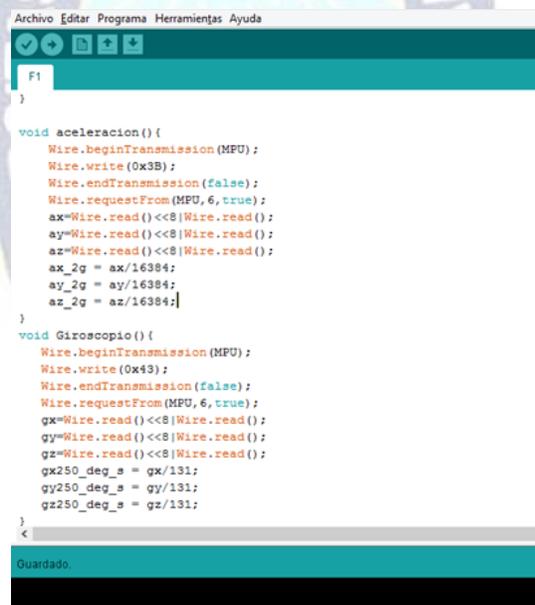
Para tal caso se debe realizar una calibración del sensor MPU6050 para tener una correlación de datos al momento de hacer las respectivas mediciones.

✓ **Sensibilidad de del sensor MPU6050.**

Según el Datasheet del MPU-6050, el valor obtenido tiene una resolución de 16 bits para cada eje de la aceleración y velocidad angular, también el rango de escala del acelerómetro es de $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$ y para el giroscopio es de $\pm 250^\circ/s$, $\pm 500^\circ/s$, $\pm 1000^\circ/s$, $\pm 2000^\circ/s$.

Se tomará la primera escala tanto para el acelerómetro y giroscopio que es $\pm 2g$ y $\pm 250^\circ/s$ respectivamente, esto debido a que la sensibilidad disminuye a medida que se aumenta a la escala más alta, eligiendo esta escala se asegura un correcto censado de datos de los diferentes ejes existentes.

En la figura 3.7 se muestra los datos obtenidos de los ejes x, y, z los cuales son convertidos a escala 2g para el acelerómetro y a escala 250 para el giroscopio.



```
Archivo Editar Programa Herramientas Ayuda
F1
}

void aceleracion() {
  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 6, true);
  ax=Wire.read()<<8|Wire.read();
  ay=Wire.read()<<8|Wire.read();
  az=Wire.read()<<8|Wire.read();
  ax_2g = ax/16384;
  ay_2g = ay/16384;
  az_2g = az/16384;
}

void Giroscopio() {
  Wire.beginTransmission(MPU);
  Wire.write(0x4B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 6, true);
  gx=Wire.read()<<8|Wire.read();
  gy=Wire.read()<<8|Wire.read();
  gz=Wire.read()<<8|Wire.read();
  qx250_deg_s = gx/131;
  qy250_deg_s = gy/131;
  qz250_deg_s = gz/131;
}
Guardado.
```

Figura 3.7. Código configuración de escala de Acelerómetro y giroscopio

Fuente: Elaboración propia

✓ **Velocidad de transmisión**

La velocidad de transmisión de datos en Arduino es bits por segundo (baudios) el cual es programable según la aplicación para la transmisión de datos en serie. Para permitir la comunicación serial se elegirá 9600 baudios.

En la figura 3.8 se muestra la respectiva configuración de la velocidad de transmisión con el cual se trabajará.

```
Archivo Editar Programa Herramientas Ayuda
F15
void setup()
{
  Wire.begin(4, 5);
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);
}
```

Figura 3.8. Velocidad de transmisión

Fuente: Elaboración propia

✓ Análisis de la aceleración y velocidad angular

Para hacer el análisis tanto para el aceleración y velocidad angular se realizó un pequeño experimento, el cual consiste en hacer caer el prototipo armada con los componentes de la fase anterior a una altura aproximadamente de un metro, entonces en esta simulación de un evento de caída se puede observar el comportamiento del acelerómetro y giroscopio, durante esta simulación se puede distinguir cuatro etapas de una caída, las cuales son ilustradas en la Figura 3.9 para el comportamiento del acelerómetro, y en la Figura 3.10 para el comportamiento del giroscopio.

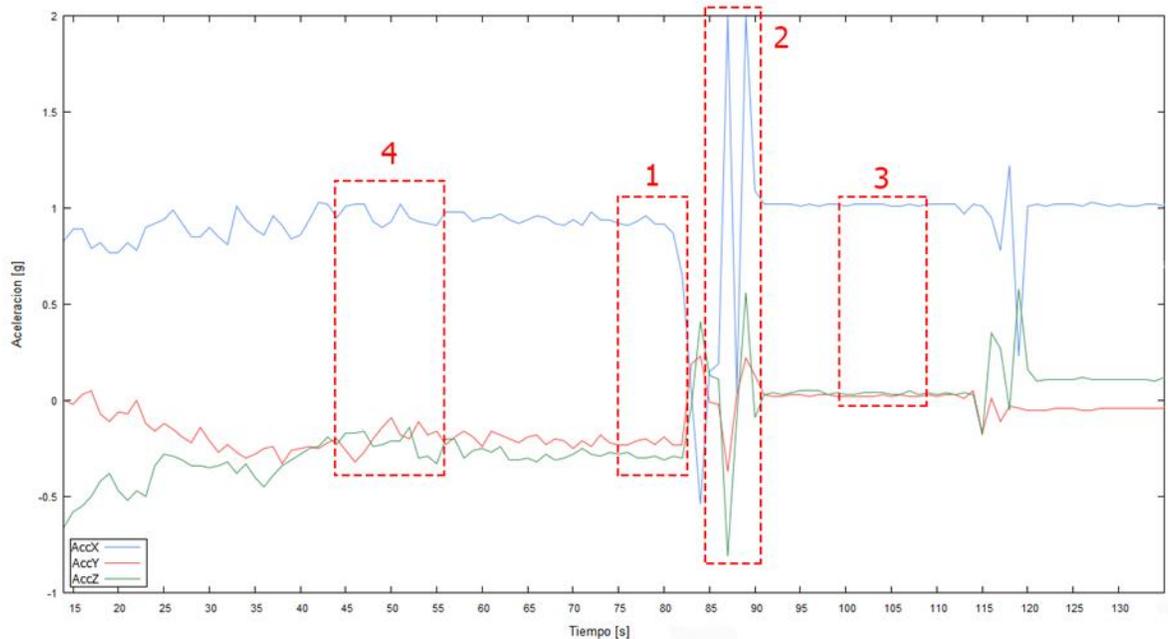


Figura 3.9. Aceleración durante una caída

Fuente: Elaboración propia

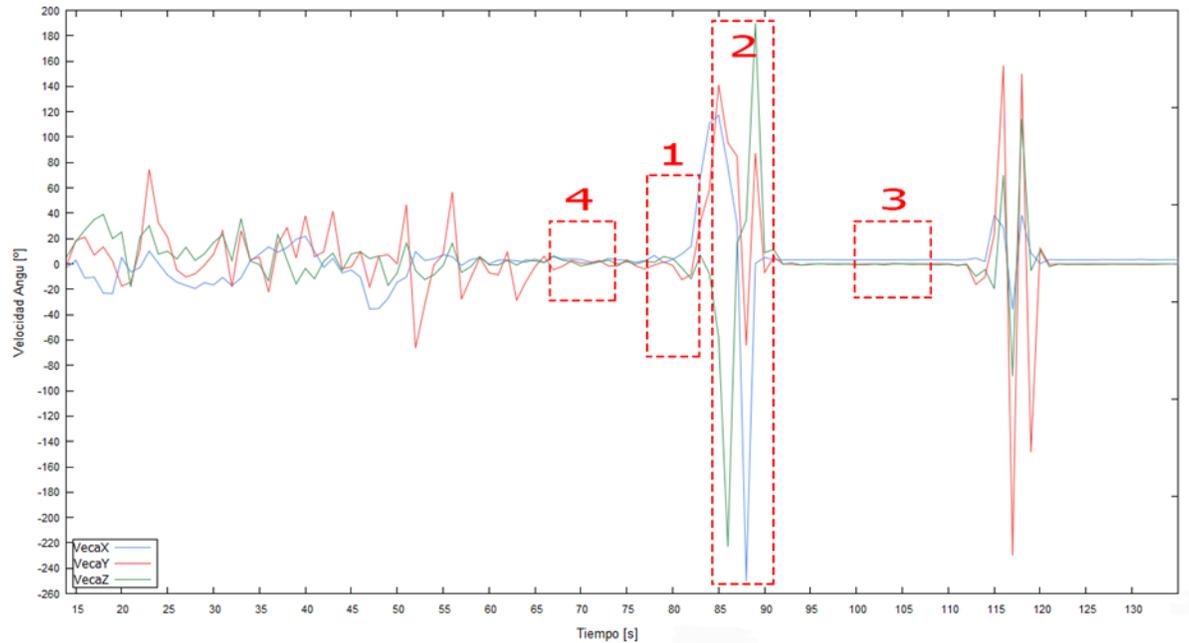


Figura 3.10. Velocidad angular durante una caída

Fuente: Elaboración propia

Las 4 etapas de una caída que se observa en los gráficos son:

1. Inicio de una caída donde el cuerpo se dirige al suelo, en el cual ocurre un efecto de ingravidez en la aceleración cambia de 1g a 0g, y la magnitud de la velocidad tiende a crecer
2. Impacto, cuando el cuerpo choco en el suelo u otros objetos aumenta un pico alto en la magnitud de la aceleración y velocidad angular.
3. Reposo, luego del impacto el cuerpo se mantiene en reposo, no suele levantarse inmediatamente, el cuerpo mantiene una posición inmóvil durante un tiempo corto.
4. Estado inicial, luego de una caída el cuerpo estará en una posición diferente al anterior en este caso al inicio de una caída, en la gráfica se observa que los ejes x y z son diferentes al comienzo de la caída.

Los valores del acelerómetro para los ejes x, y, z varían de entre 0 y 3 g, para los valores del giroscopio varían de entre 0 a 250 aproximadamente, entonces para tomar todos los valores de las aceleraciones y velocidad angular en uno solo se decide sacar la magnitud de ambos, lo cual facilita al momento de analizar los posibles eventos de caídas, como son la caída posterior, caída frontal y caídas laterales tanto izquierdo como derecho.

Para calcular la magnitud de la aceleración se aplica la siguiente formula:

$$|a| = \sqrt{a_x + a_y + a_z}$$

Donde:

a_x es aceleración en el eje x

a_y es aceleración en el eje y

a_z es aceleración en el eje z

En la Figura 3.11 se observa el código implementado en Arduino con el cual se obtiene las magnitudes totales tanto de la aceleración como la velocidad angular respectivamente.

```

Archivo  Editar  Programa  Herramientas  Ayuda
F1 $
a_abs = pow(pow((ax/16384.0),2)+pow((ay/16384.0),2)+pow((az/16384.0),2),0.5);
g_abs = sqrt(pow((gx/131.07),2)+pow((gy/131.07),2)+pow((gz/131.07),2));

```

Figura 3.11. Código de magnitudes aceleración y velocidad angular

Fuente: Elaboración propia

En las posteriores figuras se muestra diferentes gráficos en el cual se observa el comportamiento de la magnitud de la aceleración y velocidad angular en las diferentes actividades diarias, para posteriormente determinar los umbrales con los cuales se definirán los eventos de caídas.

En la figura 3.12 se muestra la gráfica de la estimación de magnitudes tanto de aceleración como de velocidad angular al experimentar una caída frontal.

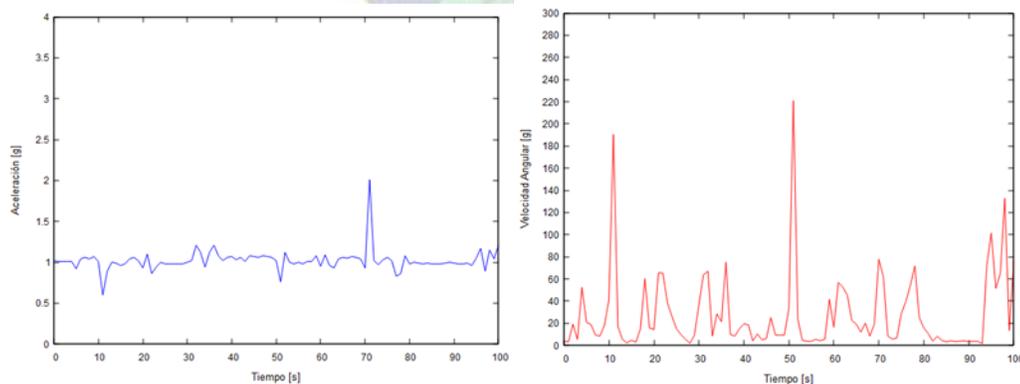


Figura 3.12. Magnitudes de caída frontal

Fuente: Elaboración propia

En la figura 3.13 se muestra la gráfica de la estimación de magnitudes tanto de aceleración como de velocidad angular al hacer la experimentación de subir las gradas.

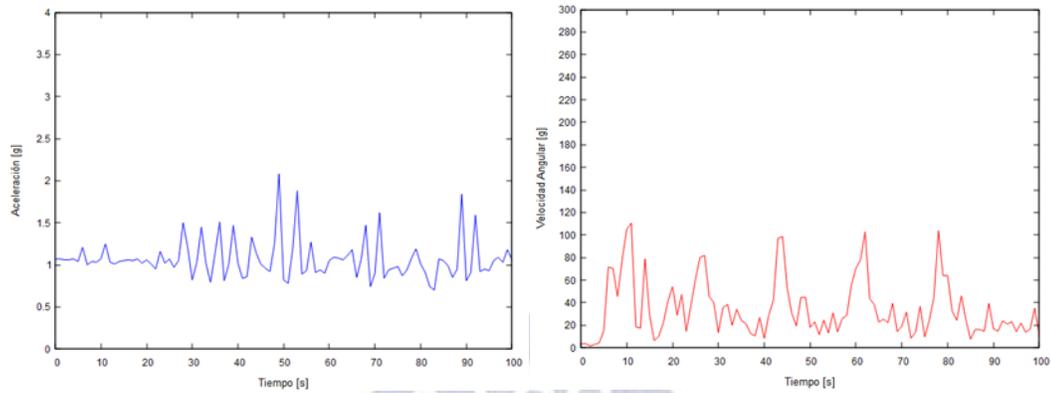


Figura 3.13. Magnitudes al subir gradas

Fuente: Elaboración propia

En la figura 3.14 se muestra la gráfica de la estimación de magnitudes tanto de aceleración como de velocidad angular al experimentar una simulación de caída lateral.

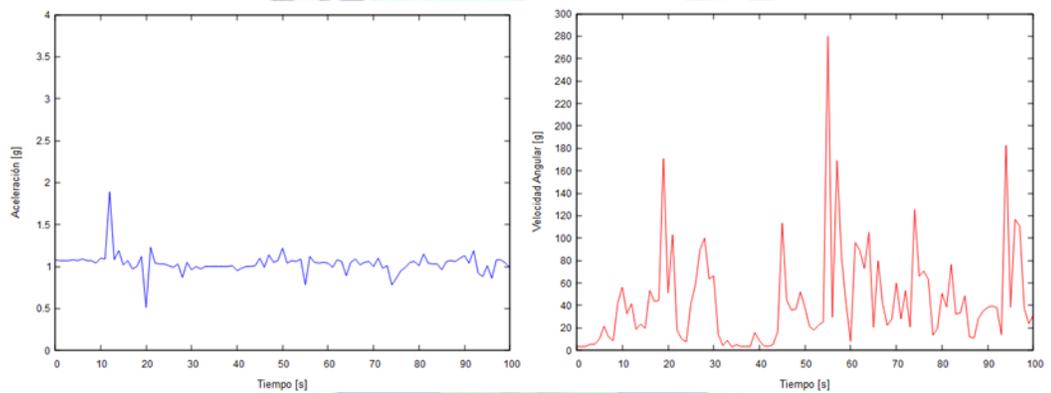


Figura 3.14. Magnitudes de caída lateral

Fuente: Elaboración propia

En la figura 3.15 se muestra la gráfica de la estimación de las magnitudes tanto de aceleración como de velocidad angular al realizar la respectiva simulación de una caída posterior.

En la figura 3.16 se muestra la gráfica de la estimación de las magnitudes tanto de aceleración como de velocidad angular al realizar la simulación de caminar en un suelo plano sin obstáculos.

En la figura 3.17 se muestra la gráfica la estimación de magnitud de la aceleración, así mismo se muestra la magnitud de la velocidad angular al realizar una simulación correspondiente de saltos.

En la figura 3.18 se muestra en la gráfica la estimación de magnitud de la aceleración como de velocidad angular al realizar la actividad de sentarse y levantarse de una silla.

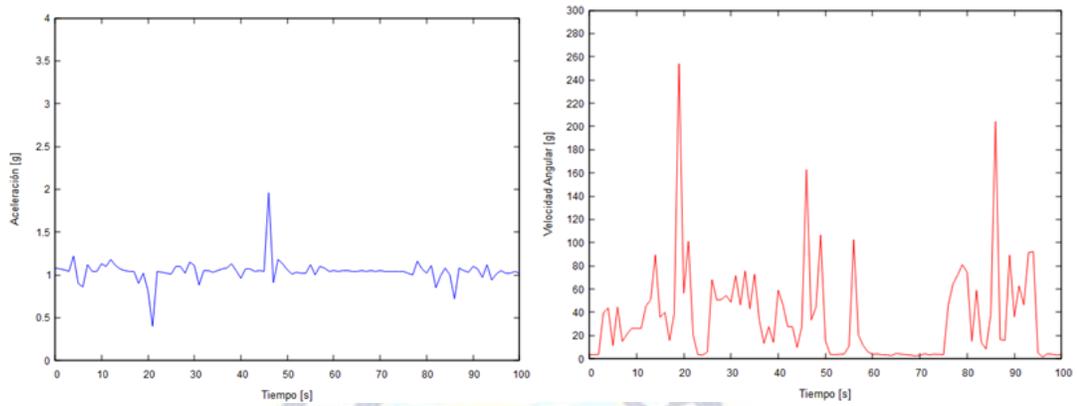


Figura 3.15. Magnitudes de caída posterior

Fuente: Elaboración propia

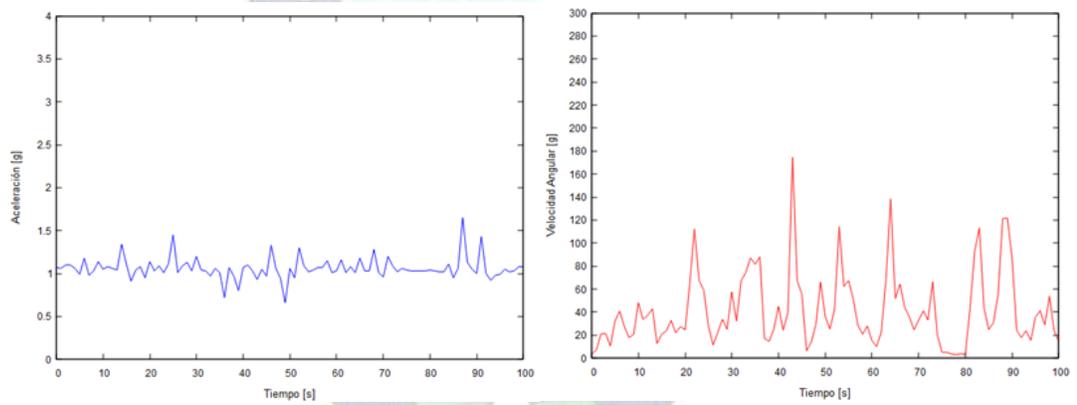


Figura 3.16. Magnitudes al caminar

Fuente: Elaboración propia

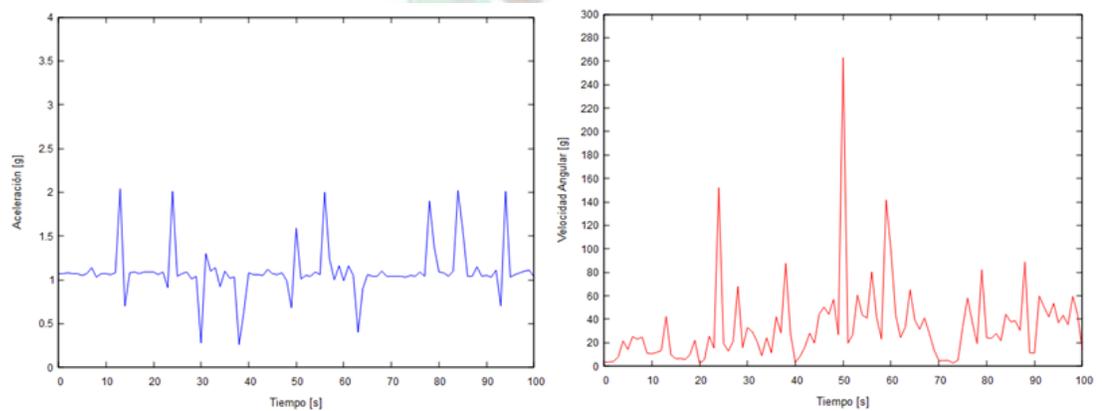


Figura 3.17. Magnitudes de hacer saltos

Fuente: Elaboración propia

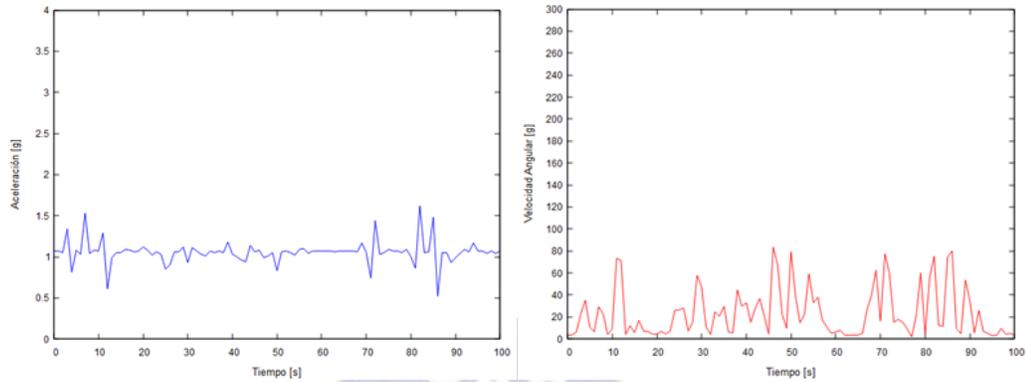


Figura 3.18. Magnitudes al momento de sentarse en una silla

Fuente: Elaboración propia

Posteriormente después de observar las gráficas se determinará un promedio de los datos obtenidos de las magnitudes de la aceleración y velocidad angular y la estimación de la orientación del cuerpo.

Al observar estas gráficas con las diferentes actividades se puede observar que el umbral superior es aproximadamente a partir de 1.5 g en promedio para las diferentes caídas en los valores de magnitud de la aceleración. Y para el giroscopio en promedio es aproximadamente a partir de 290 g/s en las diferentes caídas que existe, mientras que en las otras actividades son diferentes los valores, estos no pasan de 1g y 200g/s para la magnitud de la aceleración y la magnitud de la velocidad angular en promedio respectivamente.

✓ Orientación de la caída

Para determinar la orientación de una caída se tiene que determinar el ángulo de inclinación de la persona, La orientación de caída de una persona, depende de la posición del cuerpo dentro de un sistema de coordenadas.

En la figura 3.19 se observa a aplicación de la gravedad sobre un cuerpo con el cual podemos determinar la orientación de un cuerpo.

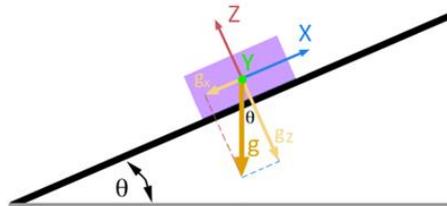


Figura 3.19. Aplicación de gravedad sobre un cuerpo

Fuente: (Llamas, 2016)

Por lo tanto, para obtener la inclinación del cuerpo del sensor MPU6050 en un sistema de tres dimensiones, se tiene las siguientes formulas con respecto al eje z, y.

$$\theta_z = \arctan\left(\frac{a_z}{\sqrt{a_y^2 + a_x^2}}\right)$$

$$\theta_y = \arctan\left(\frac{a_y}{\sqrt{a_z^2 + a_x^2}}\right)$$

Los ángulos de inclinación están en radianes entonces se harán las respectivas conversiones.

$$\left(\frac{180^0}{\text{Pi}}\right)$$

En la figura 3.20 corresponde al código implementado en Arduino para la obtención de las inclinaciones en grados con la respectiva conversión descrita anteriormente.



```

Archivo  Editar  Programa  Herramientas  Ayuda
ang_x = atan(ax / sqrt(pow(ay, 2) + pow(az, 2)))*(180.0 / 3.14);
ang_y = atan(ay / sqrt(pow(ax, 2) + pow(az, 2)))*(180.0 / 3.14);
  
```

Figura 3.20. código de obtención de ángulos en Arduino

Fuente: Elaboración propia

✓ Diseño del algoritmo de detección

El algoritmo de detección de caídas está basado en análisis de umbrales observados en distintas actividades como correr, saltar, sentarse y simulación de las diferentes caídas, con los umbrales se diferenciará entre un evento de caída con respecto a las otras actividades diarias.

Se observó durante las pruebas, cuando una persona sufre una caída la aceleración disminuye rápidamente y la velocidad angular se incrementan durante un evento de la caída. Los umbrales críticos en la aceleración y velocidad angular son utilizados para determinar si existe un evento de caída.

En la figura 3.21 se observa el diagrama de flujo en donde hace análisis de umbrales del acelerómetro, velocidad angular y posición u orientación del cuerpo.

Donde UA y UV son los umbrales de la aceleración y giroscopio respectivamente.

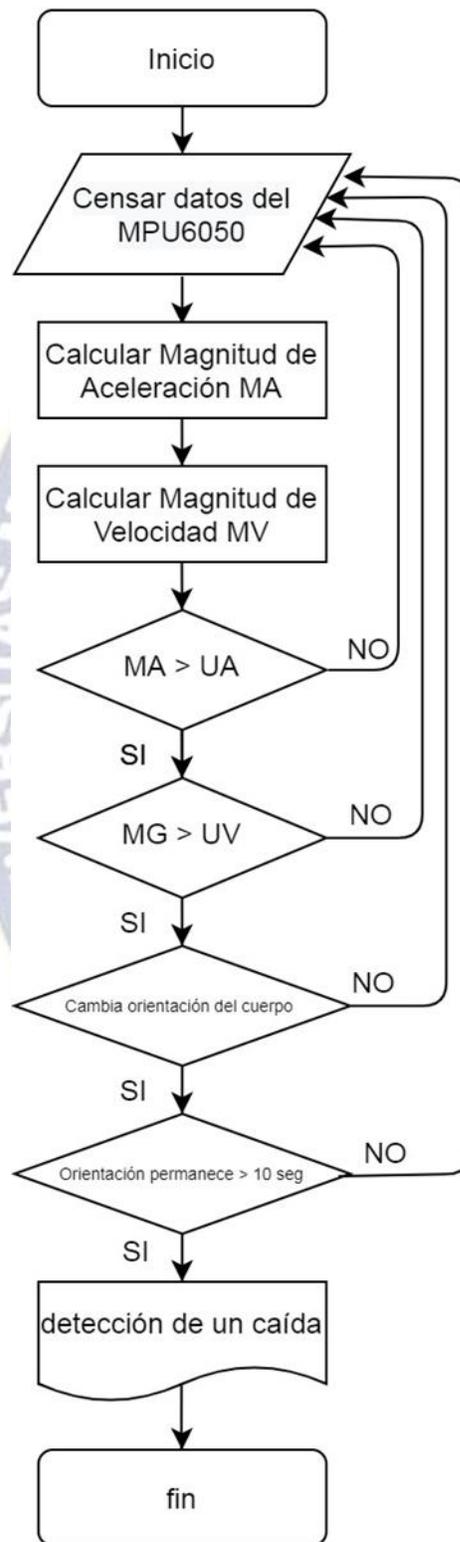


Figura 3.21. Diagrama de flujo para detectar caídas

Fuente: Elaboración propia

En la Tabla 3.11 se observa la estimación la inclinación de cuerpo en las diferentes caídas del eje Y y X respecto al eje Z, se establecen los rangos de los ángulos de inclinación en las cuatro orientaciones de las caídas.

Tabla 3.11. Estimación de inclinación del cuerpo en una caída

Tipos	Rango de inclinación
Caída frontal	[65, 90]
Caída posterior	[-55, -90]
Caída lateral derecho	[65, 90]
Caída lateral izquierdo	[-60, -90]

Fuente: Elaboración propia

✓ **Envío de notificación**

En la figura 3.22 corresponde al código de él envío de notificación, este método se llamará cuando se detecte una caída.

Para lo cual se utilizará el api de Firebase, se utilizará parámetros http de cabecera y un mensaje json el cual contiene un identificador o token lo cual es proporcionado por Firebase para enviar a un dispositivo en específico, este objeto json tiene un título y un mensaje los cuales se enviarán a la aplicación. También se realiza la especificación del método http y el servicio que se invocara que en este caso es post, se realiza la autenticación al cliente con una clave secreta lo cual se obtiene de la consola de Firebase, se especifica el tipo de contenido que se usara que en este caso es json y también se realiza la configuración correspondiente del host de Firebase.

```
String data = "{" ;
data = data + "\"to\": \"fcm.googleapis.com:9997:android:caida\"" ;
data = data + "\"notification\": {" ;
data = data + "\"titulo\": \"Deteccion de caida\"," ;
data = data + "\"mensaje\" : \"Ayuda sufrio una caida \" " ;
data = data + "}" ;
client.println("POST /fcm/send HTTP/1.1");
client.println("Authorization: key=AAAAIVT8wA:APAwh12345678901234567890");
client.println("Content-Type: application/json");
client.println("Host: fcm.googleapis.com");
client.print("Content-Length: ");
client.println(data.length());
client.print("\n");
client.print(data);
```

Figura 3.22. Código para él envío de notificaciones

Fuente: Elaboración propia

3.5.2. APLICACIÓN MÓVIL

La segunda especificación es la aplicación móvil, el cual deberá recibir las notificaciones de alertas emitidas por el NodeMCU cuando exista un evento de caída.

✓ **Notificaciones**

Primero se debe crear un proyecto siguiendo todos los pasos que indica al momento de crear un proyecto en Firebase con un nombre en específico como se muestra en la figura 3.23, en el cual se creó un proyecto con el nombre de notificaciones.

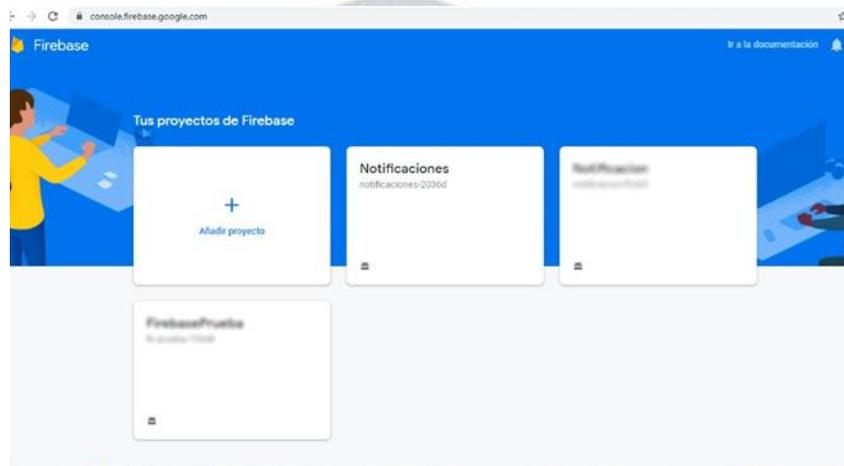


Figura 3.23. Nombre del proyecto en Consola de Firebase

Fuente: Elaboración propia

Posteriormente se procede a crear un proyecto en Android Studio, se sigue todos los pasos que indica al momento de crear como el nombre que se asigna a la aplicación, el lenguaje que se utilizara que en este caso es java, y seleccionar la versión de Android entre otras configuraciones secundarias que se realizan.

Posteriormente se tendrá una Activity en blanco una vez creada el proyecto, luego se agrega los TextView a la Activity, en el cual se colocará el mensaje de un evento de caída que se recibe, luego se da un color degradado al fondo de la Activity para que tenga una mejor vista, así también se cambia los colores al TextView para una mejor notoriedad.

Luego se agrega un GifImageView en la Activity, en el cual se coloca una imagen gif esta imagen se mostrará al momento en que llegue la notificación de caída.

Como se usará la aplicación con internet se debe dar los respectivos permisos de internet en AndroidManifest.

En la imagen 3.24 se observa el entorno de Android Studio con todos los componentes agregados y modificados que se describió anteriormente.

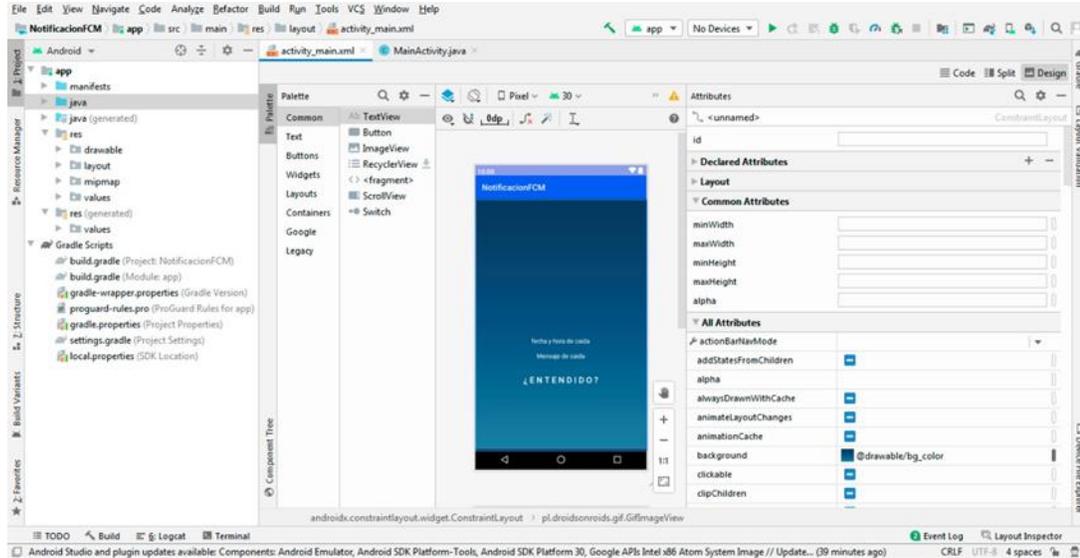


Figura 3.24. Entorno de Android Studio

Fuente: Elaboración propia

Luego se procede a configurar la aplicación Android para utilizar los servicios de Firebase, es decir añadir Firebase a la aplicación de Android, para lo cual se sigue todos los pasos que indica el asistente en la consola de Firebase, lo cual se observa en la figura 3.25.

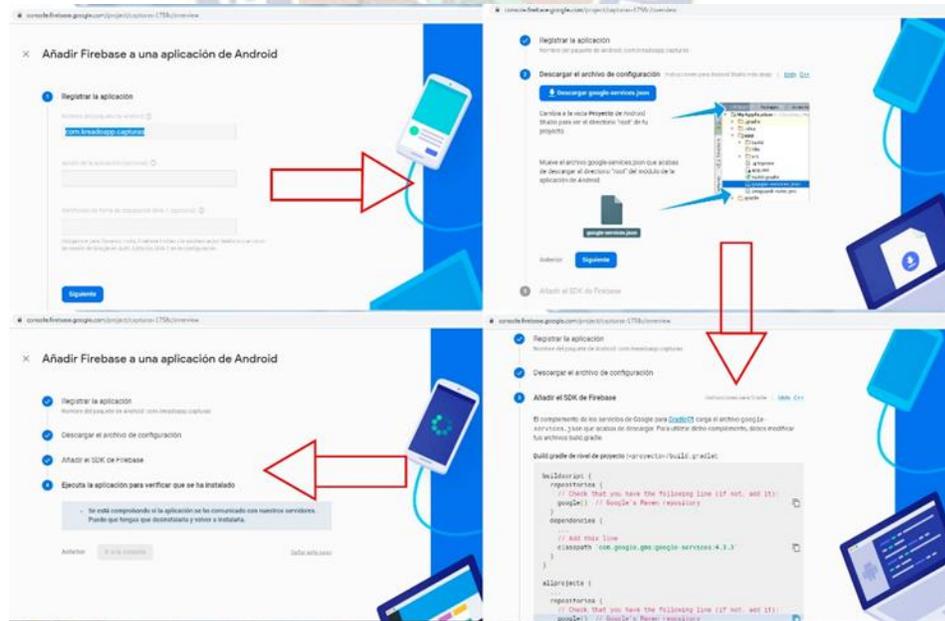


Figura 3.25. Configuración para añadir Firebase a la aplicación

Fuente: Elaboración propia

Posteriormente se crea la clase MyFirebaseMessagingService, esta clase es la encargada de recibir las notificaciones y posteriormente poderlo manejar o modificar. Cuando llegue una

notificación se verifica que no esté vacía para poderla mostrarla tal como se conoce las noticias, para lo cual se crea un método showNotification en el cual se construye la notificación, se especifica el icono, el título, el cuerpo de la notificación con los parámetros del método, hacer desaparecer cuando se presiona, asignación de un sonido que se usa cuando este activo, y al momento de tocar la notificación mandar al Activity principal tal como se muestra en la imagen 3.26.

```

public class MyFirebaseMessagingService extends FirebaseMessagingService {
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        if(remoteMessage.getData().isEmpty())
            showNotification(remoteMessage.getNotification().getTitle(),remoteMessage.getNotification().getBody());
        else showNotification(remoteMessage.getData());
    }

    private void showNotification(Map<String, String> data) {
        String title = data.get("title").toString();
        String body = data.get("body").toString();
        String NOTIFICATION_CHANNEL_ID = getString(R.string.default_notification_channel_id);
        Intent intent = new Intent( packageContext, this,MainActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        PendingIntent pendingIntent = PendingIntent.getActivity( context, this, requestCode: 0,intent,PendingIntent.FLAG_ONE_SHOT);
        Uri defaultSoundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder( context, NOTIFICATION_CHANNEL_ID);
        notificationBuilder
            .setSmallIcon(R.drawable.ic_start_ic_notification)
            .setColor(Color.rgb( red: 255, green: 160, blue: 0 ))
            .setContentTitle(title)
            .setContentText(body)
            .setAutoCancel(true)
            .setVibrate(new long[]{0,1000,500,1000})
            .setSound(defaultSoundUri)
            .setContentIntent(pendingIntent)

            .setContentInfo("info");
    }
}

```

Figura 3.26. Implementación de la clase en Android Studio

Fuente: Elaboración propia

Posteriormente se crea un método en la clase MainActivity con el cual se recibe el token, el cual es el identificador que proporciona Firebase como se ilustra en la imagen 3.27, gracias a este token se podrá enviar notificaciones al dispositivo desde Arduino.

```

FirebaseInstanceId.getInstance().getInstanceId().addOnCompleteListener(new OnCompleteListener<InstanceIdResult>() {
    @Override
    public void onComplete(@NonNull Task<InstanceIdResult> task) {
        if (!task.isSuccessful()){
            Log.w( tag: "MainActivity", msg: "error al obtener token",task.getException());
            return;
        }
        else{
            String token=task.getResult().getToken();
            Log.w( tag: "MainActivity : ",token);
        }
    }
});

```

Figura 3.27. Código para obtención de token en Android Studio

Fuente: Elaboración propia

3.6.FASE 5. TEST DE DISEÑO

En esta fase de test de diseño se verifica el correcto funcionamiento de los elementos que componen este prototipo, se documenta las verificaciones funcionales de forma individual que implica los aspectos software y hardware.

En la figura 3.28 se observa conexión física del prototipo en general con todos los componentes hardware descritos.

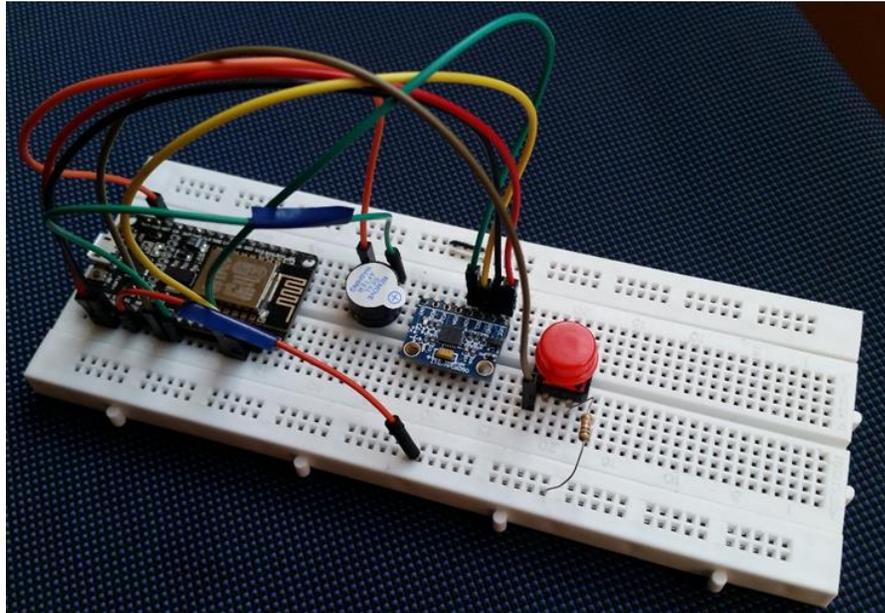


Figura 3.28. Conexión física del prototipo

Fuente: Elaboración propia

Conexión física para MPU6050

En la figura 3.29 se observa la conexión física entre el sensor MPU6050 al NodeMCU en los pines correspondientes.

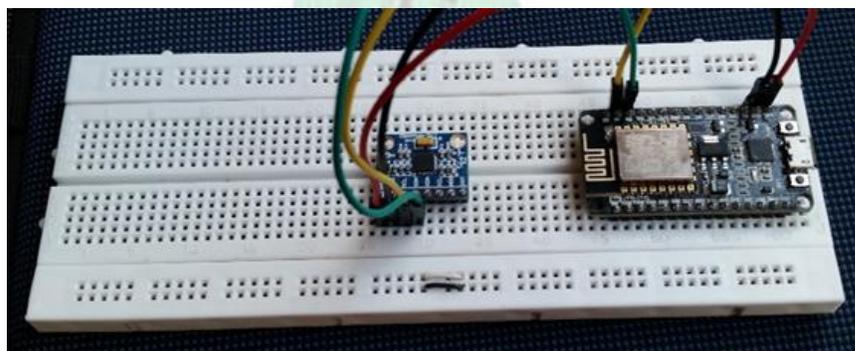


Figura 3.29. Conexión física del sensor MPU6050

Fuente: Elaboración propia

La tabla 3.12 se muestra la prueba de conexión con respecto al sensor MPU6050 y el NodeMCU.

Tabla 3.12. Prueba de conexión al sensor MPU6050

Documento de prueba 1	
Modulo	Módulo MPU6050
Actividad	Se observó gráficamente en el serial plotter y el monitor serial él envía de los datos tanto del acelerómetro como del giroscopio.
Observación	La conexión respondió correctamente.
Detalle de la observación	Algunos pines de conexión con este módulo suelen variar, para lo cual se debe hacer una correcta elección del pin del NodeMCU.

Fuente: Elaboración propia

Conexión física para zumbador y pulsador

En la figura 3.30 se observa la correspondiente conexión física entre el sensor zumbador el pulsador al NodeMCU.

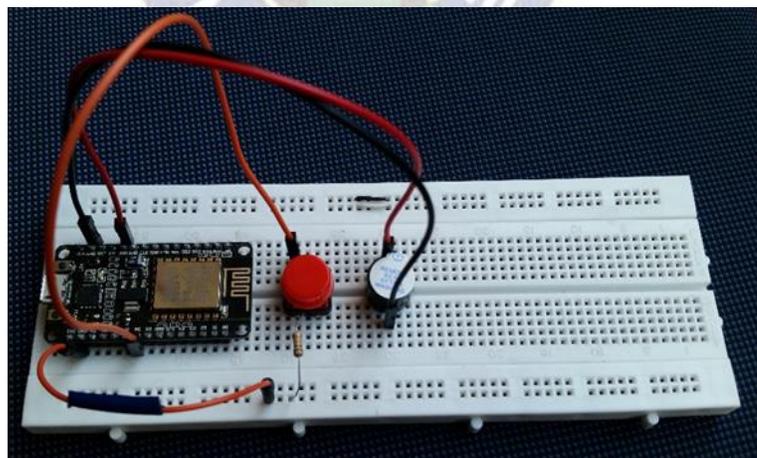


Figura 3.30. Conexión física del zumbador

Fuente: Elaboración propia

La tabla 3.13 se muestra la prueba de conexión con respecto a los sensores pulsador y zumbador al NodeMCU.

Tabla 3.13. Prueba de conexión al sensor pulsador y zumbador

Documento de prueba 2	
Módulos	pulsador y zumbador
Actividad	Se observó en el monitor serial el correcto funcionamiento tanto del zumbador y el pulsador al momento de pulsar y de mandar señales para que se active el zumbador por un pequeño lapso de tiempo.
Observación	La conexión respondió correctamente.
Detalle de la observación	Por lo general estos componentes son compatibles con las mayoría de las placas lo que lo hace menos dificultoso.

Fuente: Elaboración propia

En la figura 3.31 se observa la conexión de Arduino a Firebase y la recepción de una prueba de notificación de la aplicación desde Firebase.

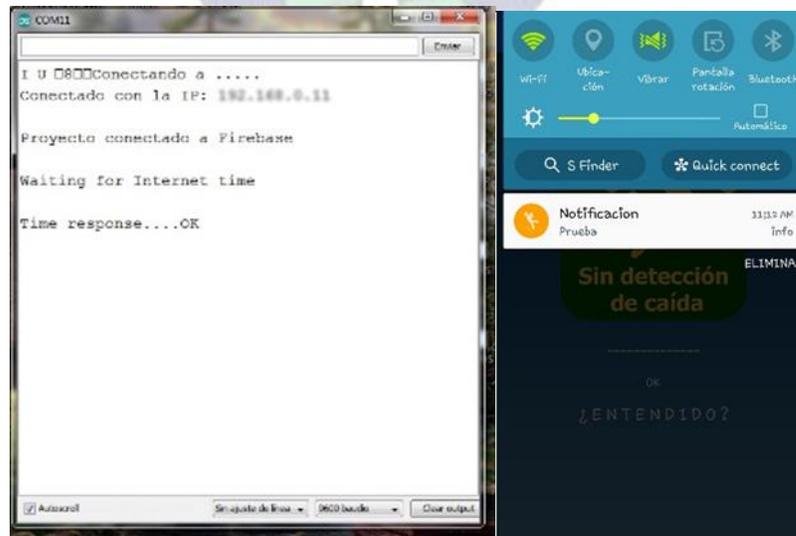


Figura 3.31. Conexión a Firebase y recepción de notificación en la aplicación

Fuente: Elaboración propia

La tabla 3.14 se muestra la prueba de conexión a Firebase desde Arduino y él envió de datos desde Firebase a la aplicación.

Tabla 3.14. Prueba de conexión al servidor

Documento de prueba 3	
Módulos	Conexión
Actividad	Se observó la respectiva conexión del NodeMCU a Firebase, así también él envió de una prueba de notificación desde Firebase a la aplicación.
Observación	La conexión respondió correctamente.
Detalle de la observación	Él envió de datos hacia la aplicación dependerá de la velocidad del internet del que se tenga.

Fuente: Elaboración propia

3.7.FASE 6. TEST FUNCIONAL

En esta fase de prueba de las funcionalidades se afirma la integración de cada unidad que forma el prototipo verificada en la fase anterior de los diferentes módulos, así también esta fase nos permite verificar los objetivos que se planteó.

En la figura 3.32 se observa la unión de todos los componentes que forma parte del prototipo detector de caídas y así también se muestra la aplicación en donde se enviara las notificaciones de alertas ante un evento de caída.



Figura 3.32. Prototipo final

Fuente: Elaboración propia

Una vez que se integran todos los módulos se realiza la verificación del funcionamiento del prototipo el cual permitirá detectar un evento de caída lo cual se muestra en la tabla 3.15 donde el prototipo detecta los diferentes tipos de caídas en distintas situaciones.

En esta fase las pruebas fueron realizadas por una sola persona a fin de probar el funcionamiento del prototipo, los eventos de caídas que fueron simulados son caídas frontal, posterior, lateral derecho, lateral izquierdo, para eso se utilizó una colchoneta la cual absorbe gran parte del impacto, no se realizó directamente en un suelo rígido.

Tabla 3.15. Pruebas de caídas

Tipos	Acertadas	No Acertadas	Cantidad Total De Pruebas
Caída frontal	2	1	3
Caída posterior	3	0	3
Caída lateral derecho	2	1	3
Caída lateral izquierdo	2	1	3

Fuente: Elaboración propia

A continuación, se observa las gráficas de experimentos de caídas realizadas por una sola persona en una colchoneta, en la figura 3.33 se muestra al usuario probando el prototipo.



Figura 3.33. prueba del prototipo

Fuente: Elaboración propia

Caída de frente. La persona se encuentra en posición vertical, luego de esa posición se simula el evento de caída hacia adelante, en la figura 3.34 se observa el cambio de aceleración y velocidad.

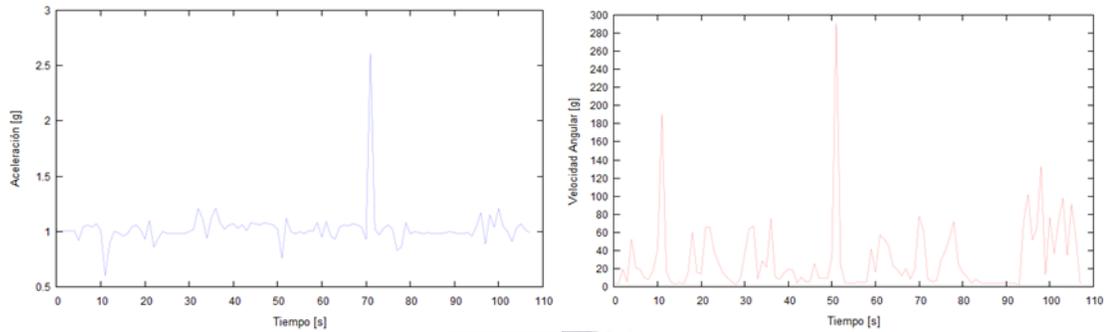


Figura 3.34. Magnitud de la aceleración y velocidad producida al caer de frente

Fuente: Elaboración propia.

Caída de espalda. La persona se encuentra en posición vertical, luego de la señal deja caer el cuerpo hacia atrás, en la figura 3.35 se observa el cambio de aceleración y velocidad respecto a la caída producida.

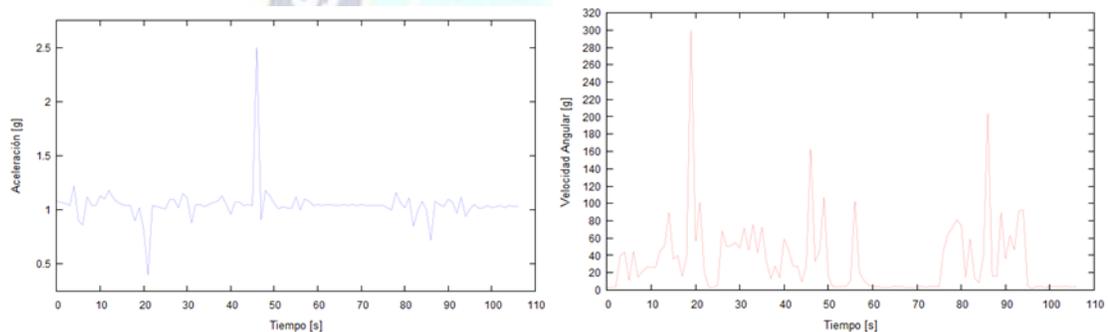


Figura 3.35. Magnitud de la aceleración y velocidad al caer de espalda

Fuente: Elaboración propia.

Caída lateral derecho. La persona se encuentra de pie, luego de la señal deja caer el cuerpo hacia lado derecho, en la figura 3.36 se observa el cambio de aceleración y velocidad con respecto a la caída.

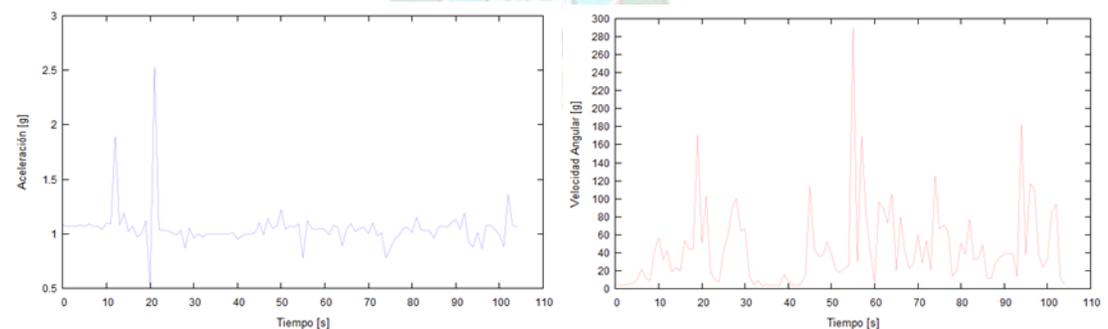


Figura 3.36. Magnitud de aceleración y velocidad de una caída lateral derecho

Fuente: Elaboración propia.

Caída lateral izquierdo. La persona se encuentra de pie, luego de la señal deja caer el cuerpo hacia lado izquierdo, en la figura 3.37 se observa el cambio de aceleración y velocidad producidas al sufrir una caída.

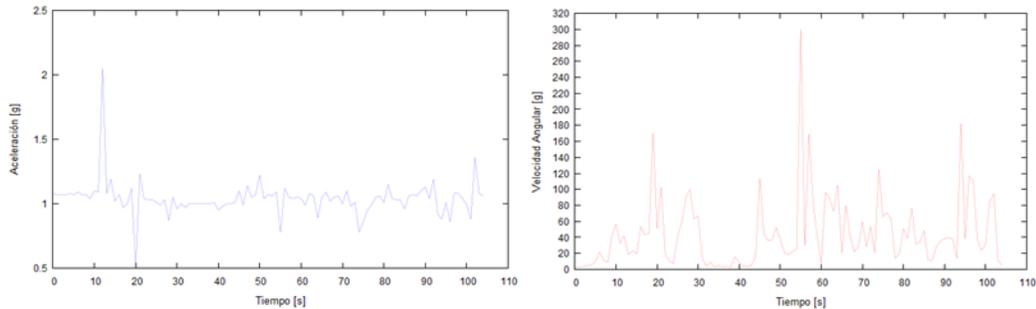


Figura 3.37. Magnitud de aceleración y velocidad de una caída lateral izquierdo

Fuente: Elaboración propia.

3.8. FASE 7. TEST DE ESPECIFICACIONES

En esta fase se debe realizar la instalación del prototipo en una ubicación final para realizar las pruebas ante un escenario real.

La ubicación final se observa en la figura 3.38 donde ira puesta el prototipo detector de caída es en la cintura debido a que no existe tanto movimiento como en otras partes del cuerpo el cual generaría falsas alarmas debido a los movimientos que hace una persona en sus actividades cotidianas.



Figura 3.38. Ubicación del prototipo y prototipo final

Fuente: Elaboración propia

Los eventos de caídas fueron simulados por personas adultas de entre 24 a 30 años el cual se detalla la tabla 3.16 las cuales simularon las caídas frontal, posterior, lateral derecho,

lateral izquierdo, para tal caso se utilizó una colchoneta la cual absorbe gran parte del impacto, cada prueba dura alrededor de 3 a 5 minutos aproximadamente y las pruebas son observadas y registradas.

Tabla 3.16. Personas de prueba

Personas	Edad	Sexo	Cantidad de pruebas
x	30	M	7
x	25	F	5
x	26	M	6
x	27	M	4
x	27	M	6
x	28	M	5
x	24	F	4
x	27	M	5
x	27	M	7
x	29	M	5

Fuente: Elaboración propia

En la tabla 3.17 se muestra la cantidad de caídas que simularon con el fin de verificar si efectivamente hace la detección de un evento de caída así también se observa la cantidad de caídas acertadas, no acertadas de las caídas simuladas.

Tabla 3.17. Pruebas de caídas

Tipos	Acertadas	No Acertadas	Cantidad Toral De Pruebas
Caída frontal	10	0	10
Caída posterior	14	1	15
Caída lateral derecho	9	5	14
Caída lateral izquierdo	10	5	15

Fuente: Elaboración propia

También se observó que hace él envió correcto de alarmas a la aplicación cuando sucede una caída y la aplicación hace la recepción correctamente de las notificaciones.

4. EVALUACIÓN DE RESULTADOS

4.1. INTRODUCCIÓN

En el presente capítulo tiene como objetivo demostrar la hipótesis planteada en el primer capítulo la validez y veracidad de la hipótesis, para lo cual se hará el uso de un método estadístico para determinar su aceptación o rechazo.

La prueba se realizará en base al resultado y análisis de datos obtenidos en la fase de test de especificaciones del modelo en V.

4.2. PRUEBA DE HIPÓTESIS

Los datos a ser tomados para detectar una caída son: detectar caídas frontales, posteriores, laterales izquierdo y derecho.

Para la detección de las caídas se realizaron una serie de pruebas con todas las medidas necesarias la cual se refleja en la fase de test del modelo en V.

Para hacer la demostración de la hipótesis primero se establece la hipótesis H_0 y H_1 hipótesis nula e hipótesis alternativa respectivamente para la evaluación.

Hipótesis nula:

H_0 = Mediante un prototipo para la detección de caídas basado en Arduino e Intente de las Cosas no permite detectar las caídas laterales, frontal y posterior en una persona adulto mayor.

Hipótesis alternativa:

H_1 = Mediante un prototipo para la detección de caídas basado en Arduino e Intente de las Cosas se permite detectar las caídas laterales, frontal y posterior en una persona adulto mayor.

Posterior mente se define el nivel de significancia, que se interpreta como el error que se comete al rechazar la hipótesis nula H_0 .

Para este análisis, el nivel de significancia que se tomará será de 0,05.

Nivel de significancia = 0.05

El estadístico de prueba que se empleará en la Chi-Cuadrada para todas las celdas, mismo que permitirá determinar la validez de la hipótesis utilizando la siguiente formula.

$$x_{\text{calc}}^2 = \sum \frac{(f_0 - f_e)}{f_e}$$

Donde:

f_0 = Frecuencia del valor observado.

f_e = Frecuencia del valor esperado.

x_{cal}^2 = Chi – Cuadrado calculado.

En la tabla 4.1 se observa el resumen de los datos obtenidos de las pruebas realizadas en los diferentes tipos de caídas.

Tabla 4.1. Valores observados de la detección de caídas

Tipos	Detectada	No detectada	Total
Caída frontal	10	0	10
Caída posterior	14	1	15
Caída lateral derecho	9	5	14
Caída lateral izquierdo	10	5	15
Total	43	11	54

Fuente: Elaboración propia

A partir de la tabla 4.1 de los valores observados, se procede a calcular el valor o frecuencia esperada, los cuales se calculan de la siguiente manera.

$$f_e = \frac{\text{total columna} * \text{total fila}}{\text{suma total}}$$

En la tabla 4.2 se observa el cálculo de los valores esperados con respecto a los valores de la tabla anterior.

Tabla 4.2. Resultado de valor esperado

f_e	f_e
$\frac{43 * 10}{54} = \frac{215}{27}$	$\frac{11 * 10}{54} = \frac{55}{27}$
$\frac{43 * 15}{54} = \frac{215}{18}$	$\frac{11 * 15}{54} = \frac{55}{18}$
$\frac{43 * 14}{54} = \frac{301}{27}$	$\frac{11 * 15}{54} = \frac{77}{27}$
$\frac{43 * 15}{54} = \frac{215}{18}$	$\frac{11 * 15}{54} = \frac{55}{18}$

Fuente: Elaboración propia

El resumen de los datos obtenidos del valor o frecuencia esperada se detalla en la tabla 4.3.

Tabla 4.3. Valores esperados

Tipos	Detectada	No detectada	Total
Caída frontal	7.96	2.04	10
Caída posterior	11.94	3.06	15
Caída lateral derecho	11.14	2.84	14
Caída lateral izquierdo	11.94	3.06	15
Total	43	11	54

Fuente: Elaboración propia

Una vez que se tiene calculado los valores observados y los valores esperados de las respectivas tablas, se procede a calcular el valor de la Chi-Cuadrado, empleando la fórmula del estadístico de prueba mencionada anteriormente.

$$x_{\text{calc}}^2 = \frac{(10 - 7.96)^2}{7.96} + \frac{(0 - 2.04)^2}{2.04} + \frac{(14 - 11.94)^2}{11.94} + \frac{(1 - 3.06)^2}{3.06} + \frac{(9 - 11.14)^2}{11.14} + \frac{(5 - 2.84)^2}{2.84} + \frac{(10 - 11.94)^2}{11.94} + \frac{(5 - 3.06)^2}{3.06} = 7.8806$$

$$x_{\text{calc}}^2 = 7.8806$$

Posteriormente se debe calcular el Chi-Cuadrado crítico, para lo cual se debe determinar los grados de libertad lo cual se calcula de la siguiente forma:

$$Gl = (N^0 \text{ de filas} - 1) * (N^0 \text{ de columnas} - 1)$$

Reemplazando los valores, como la tabla tiene 4 filas y 2 columnas se tiene el valor de 3 como se observa a continuación.

$$Gl = (4 - 1) * (2 - 1)$$

$$Gl = 3$$

El valor del Chi-Cuadrado crítico tiene grado de libertad de 3 y un nivel de significancia de 0.05, con estos valores se hace el cálculo de la distribución de Chi-cuadrado crítico, los cuales se obtienen de la tabla el valor correspondiente y por lo cual se obtiene el siguiente resultado.

$$x_{\text{critico}}^2 = 7.815$$

$$X_{\text{calculado}}^2 = 7.8806 < X_{\text{critico}}^2 = 7.815$$

4.3. ANÁLISIS DE RESULTADOS

Dado que 7.8806 es mayor que el valor crítico de 7.815 obtenido con grado de libertad de 3 y el margen de error de 0.05 en la tabla del Chi-Cuadrado, cae en la región de rechazo como se observa en la figura 4.1, por cual se llega a rechazar la hipótesis nula “Mediante un prototipo para la detección de caídas basado en Arduino e Intente de las Cosas no permite detectar las caídas laterales, frontal y posterior en una persona adulto mayor”.

Por tanto, se acepta la hipótesis alternativa “El uso de Arduino e IoT permite al prototipo detectar las caídas laterales, posterior y frontal en una persona adulto mayor”.

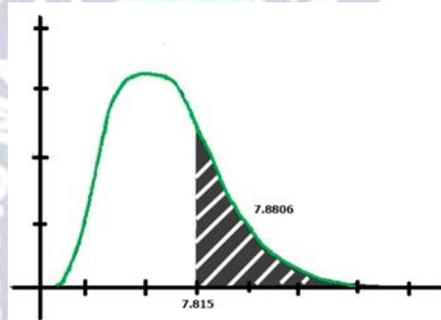


Figura 4.1. Aceptación de la hipótesis

Fuente: Elaboración propia

5. CONCLUSIONES Y RECOMENDACIONES

5.1. INTRODUCCIÓN

En el presente capítulo se mostrarán las conclusiones y recomendaciones a las cuales se llegaron, el cumplimiento del objetivo general, objetivos específicos y las recomendaciones correspondientes.

5.2. CONCLUSIONES

El presente trabajo de investigación emboca en el diseño e implementación de un prototipo detector de caídas con la ayuda de diferentes tecnologías Arduino e IoT, se logró integrar los componentes en la parte física como la placa NodeMCU, sensor MPU-6050, zumbador, pulsador. La parte de software se logró integrar Arduino con Firebase y Android Studio para la comunicación con el hardware.

El objetivo planteado en el capítulo uno “Desarrollar un prototipo portable mediante la plataforma Arduino e Internet de las Cosas para la detección de las caídas en adultos mayores” se cumplió con el diseño, implementación y pruebas al prototipo desarrollado en función a la metodología en V.

Los objetivos específicos fueron cumplidos:

- “Ayudar a los adultos mayores a minimizar los daños físicos, consecuencias en la salud causadas por una caída”: al hacer el uso del prototipo se logra disminuir los daños físicos debido a que el prototipo detecta las caídas e informa este tipo de evento para su pronta asistencia y así evitar futuros problemas agravantes.
- “Notificar al responsable del adulto mayor el evento de una caída, de tal manera disminuya el tiempo de asistencia”, se desarrolló una aplicación para un dispositivo móvil en donde el encargado recibe las notificaciones de un evento de caída que sufre el portador del dispositivo, por otro lado, también incorpora una alarma sonora con los cuales ayudan a disminuir el tiempo de asistencia ante un evento de caída.
- “Minimizar la mortalidad ante un evento de caída mediante un aviso (alarma sonora e inalámbrica) en un determinado tiempo”, se implementó en el prototipo una alarma sonora e inalámbrica, los cuales son activados en cuanto detecta un evento de caída.

- “Hacer uso de dispositivos electrónicos que van apegados al cuerpo y este dispositivo va ligado a una aplicación, el cual ofrece respuestas inmediatas, para acudir a la atención en caso de una caída”, se desarrolló el prototipo el cual está compuesto por los componentes hardware tales como una placa de desarrollo NodeMCU el cual se encarga de procesar la información, un sensor inercial mpu-6050 el cual proporciona datos tales como el aceleración y velocidad de los tres ejes, un botón para cancelar las falsas alarmas, un Buzzer para la alarma sonora y así mismo se desarrolló un aplicación en donde se recibe las notificaciones de alarma de caídas, este prototipo es portable debido al tamaño el cual va puesto en la parte de la cintura de la persona para detectar caídas.
- “Hacer el uso de tecnologías libre para el desarrollo del prototipo”, se desarrolló un prototipo utilizando software y hardware libre lo cual es relativamente económico y accesible el desarrollo lo cual se detalla en la tabla 5.1 y 5.2.
- “Realizar las pruebas de funcionamiento del prototipo”, se realizó las respectivas pruebas como se muestra en el capítulo cuatro en la fase siete del modelo en V.

El costo del material empleado en el prototipo se detalla en la siguiente tabla 5.1.

Tabla 5.1. Cantidad y costo total de materiales utilizados

Nro.	Componente	Cantidad	Costo Bs
1	Placa NodeMCU	1	40
2	Sensor MPU6050	1	30
3	Batería	1	40
4	Zumbador	1	5
5	Botón	1	2
6	Cables	20	6
7	Resistencia	4	2
8	Placa de baquelita	1	10
Total			140

Fuente: Elaboración propia

Durante el desarrollo del prototipo detector de caídas se utilizó varios tipos de software, pero todos estos fueron de libre acceso, por lo cual no se tuvo algún costo en el software lo cual se detalla en la tabla 5.2.

Tabla 5.2. Detalle de costos de software

Nro.	Programa	descripción	Costo Bs
1	java	Lenguaje de programación	0
2	C++	Lenguaje de programación	0
3	IDE Arduino	Entorno de desarrollo	0
4	IDE Android Studio	Entorno de desarrollo	0
5	Firebase	Plataforma de desarrollo en la nube	0
Total			0

Fuente: Elaboración propia

La hipótesis planteada en el presente trabajo sostiene que “Mediante un prototipo para la detección de caídas basado en Arduino e Intente de las Cosas se permite detectar las caídas laterales, frontal y posterior en una persona adulto mayor”, donde en el capítulo cuatro se puede verificar la demostración de la hipótesis el cual respalda al prototipo detectar las caídas existentes en una persona portador del dispositivo.

5.3. RECOMENDACIONES

Las recomendaciones de la presente tesis son:

- Utilizar las librerías correspondientes para la placa NodeMCU, sensor mpu-6050, debido a que al importar una librería incorrecta provoca que no recolecte los datos correctos o tengan fallas.
- Al momento de programar en el NodeMCU tomar en cuenta los puertos de comunicación a ser utilizados, debido a que si no reconoce el puerto no será posible cargar el código.
- Considerar el voltaje de alimentación del hardware como es la placa NodeMCU, sensor mpu-6050 y de más, debido a que una mala alimentación afectaría al funcionamiento e incluso llegando a quemar componentes de la placa.
- En cuanto a la aplicación se recomienda tener una buena conexión de internet para recibir las notificaciones de alarma ante un evento de caída.

5.4. RECOMENDACIONES FUTURAS

- En cuanto a los trabajos posteriores se recomienda a mejorar el prototipo con la implementación de otros módulos como, por ejemplo, sensor de ritmo cardiaco, temperatura y otros para tener un mejor control de la persona adulta mayor orientadas hacia la telemedicina.
- Para una mejor detección de caídas se recomienda tener más variables para censar, se podría incorporar sensores de sonido, sensores de impacto, sensores de inclinación entre otros.
- Para una mejor detección aplicar algoritmos predictivos para la detección de caídas.
- Se recomienda a mejorar el prototipo en cuanto al tamaño del prototipo para una mejor portabilidad.



BIBLIOGRAFÍA

- Analog Devices. (2013). *Tecnologías de sensores para acelerómetros*. Obtenido de Digi-Key: <https://www.digikey.com/en/articles/sensor-technologies-for-accelerometers>
- Andalucía es Digital. (2016). *¿qué es internet de las cosas y cómo va a cambiar nuestra sociedad?* Obtenido de Andalucía es Digital: <https://www.blog.andaluciaesdigital.es/que-es-internet-de-las-cosas/>
- Arduino. (2016). *Interfaz del sensor de giroscopio de 3 ejes BMG160 con Arduino*. Obtenido de Arduino Project Hub: <https://create.arduino.cc/projecthub/Dcube/interfacing-of-3-axis-gyroscope-sensor-bmg160-with-arduino-1c75b6>
- Arduino. (2020). *¿Por qué Arduino?* Obtenido de Arduino.
- Barrera, A. (2016). *¿Qué es y para qué sirve?* Obtenido de Nest_u: <https://www.nextu.com/blog/que-es-json/>
- Camacho, C. (2019). *El ejercicio previene el riesgo de caídas en adultos mayores*. Obtenido de Los Tiempos: <https://www.lostiempos.com/tendencias/salud/20191007/ejercicio-previene-riesgo-caidas-adultos-mayores>
- Cárdenas, A. (2016). *Secmotic Innovation*. Obtenido de ¿Qué es una plataforma IoT?: <https://secmotic.com/plataforma-iot/>
- Carnuccio, E., Valiente, W., Volker, M., De Luca, G., Garcia, G., Giulianelli, D., & Barillaro, S. (2017). *Desarrollo de un Prototipo detector de caídas utilizandola la placa Intel Galileo Generación I y el sensor MPU6050*. Universidad Nacional de La Matanza.
- Christensson, P. (2013). *Definición NoSQL*. Obtenido de TechTerms: <https://techterms.com/definition/nosql>
- Crespo, E. (2019). *Sensores Arduino*. Obtenido de Aprendiendo Arduino: <https://aprendiendoarduino.wordpress.com/2018/04/14/sensores-arduino-3/#comment-19063>
- Cruz, E., González, M., López, M., Godoy, I., & Ulises Pérez, M. (2014). *Caídas: revisión de nuevos conceptos*. *Revista HUPE*, 13, 2. Obtenido de http://revista.hupe.uerj.br/detalhe_artigo.asp?id=476

- del Valle Hernández, L. (2017). *Aplicaciones del IoT usos prácticos en el mundo real*. Obtenido de ProgramarFacil: <https://programarfacil.com/podcast/aplicaciones-del-iot-reales/>
- Dub, A. (2015). *Arduino, ADXL335 Sensor*. Obtenido de Alonsodub: <https://alonsodub.wordpress.com/2012/09/18/adxl335-acelerometro-eje-x-y-z/>
- Estrada. (2017). *Qué es Arduino – Introducción*. Obtenido de Hetpro: <https://hetpro-store.com/TUTORIALES/que-es-arduino/>
- Estrada, s. (2014). *Uso y Aplicaciones con Arduino*. Obtenido de Prezi: https://prezi.com/ih2_mpb3-hre/arduino-es-una-plataforma-de-hardware-de-codigo-abierto-bas/
- Ferreño, E. (2020). *Esto es lo que puedes hacer con este programa*. Obtenido de El androide libre: <https://elandroidelibre.elespanol.com/2020/01/android-studio-que-es-y-para-que-se-utiliza.html>
- Garcia, R. (2017). *INE: Población adulto mayor llegó a un millón y 67% está en el eje central de Bolivia*. Obtenido de kawsachun Coca Red Multimedia: <http://www.kawsachuncoca.com/index.php/2012-12-24-07-26-10/item/263-ine-poblacion-adulto-mayor-llego-a-un-millon-y-67-esta-en-el-eje-central-de-bolivia>
- García, V. (2018). *Configurar el MPU6050*. Obtenido de Electronica practica aplicada: <https://www.diarioelectronicohoy.com/blog/configurar-el-mpu6050>
- González Salgueiro, L., Areces Gonzalez, A., & Rueda Rodríguez, P. (2018). *Help Button for Elderly People on the Arduino Platform*. Universidad de Pinar del Río, Cuba. doi:10.18046/syt.v16i45.2748
- González, L. J., Areces, A., & Rueda, P. A. (2018). *Help Button for Elderly People on the Arduino Platform*. Universidad de Pinar del Río, Cuba. doi:10.18046/syt.v16i45.2748
- Goodrich, R. (2013). *Acelerómetros: qué son y cómo funcionan*. Obtenido de LiveScience : <https://www.livescience.com/40102-accelerometers.html>
- Goodrich, R. (2018). *Acelerómetro vs.GiroscoPIO: ¿Cuál es la diferencia?* Obtenido de Livescience: <https://www.livescience.com/40103-accelerometer-vs-gyroscope.html>
- Haro, J. (2015). *MMA7361 Sensor Acelerometro*. Obtenido de Hetpro: <https://hetpro-store.com/TUTORIALES/mma7361-sensor->

acelerometro/#:~:text=MMA7361%20Sensor%20Acelerometro%20%E2%80%93%20Introducci%C3%B3n,con%20respecto%20al%20eje%20terrestre.

Hernández, L. (2016). *ProgramarFacil*. Obtenido de Proyectos IoT con Arduino, las plataformas más importantes: <https://programarfacil.com/podcast/proyectos-iot-con-arduino/>

Hernández, L. (2018). *ProgramarFacil*. Obtenido de NodeMCU tutorial paso a paso desde cero: <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>

Holloway, C. (2018). *Itmasters*. Obtenido de Plataformas IoT: Qué son y cómo elegir la mejor para el negocio: <https://itmastersmag.com/noticias-analisis/plataformas-iot-que-son-y-como-elegir-la-mejor-para-el-negocio/>

INE. (2016 - 2017). *Encuesta de Hogares* . Obtenido de Catalogos ANDA: <http://anda.ine.gob.bo/index.php/catalog/54>

INE. (26 de Agosto de 2017). *Población adulta mayor boliviana tiende a incrementarse en los próximos años*. Obtenido de Instituto Nacional de Estadística: <https://www.ine.gob.bo/index.php/prensa/notas-de-prensa/item/1419-poblacion-adulta-mayor-boliviana-tiende-a-incrementarse-en-los-proximos-anos>

InvenSense. (2011). *ITG-3200 Product Specification*. Obtenido de InvenSense : http://www.farnell.com/datasheets/2645523.pdf?_ga=2.66550190.1654861532.1600382244-183006703.1600382244

Lanzarini, L., & Ridolfi, P. (2017). *Prototipo funcional de un sistema de detección de caídas basado en la plataforma CIAA*. Universidad Nacional de La Plata.

Lariza , S., & Perez, M. (2012). *Modelo en V*. Obtenido de Ingeniería de Software: <http://softwareverde.blogspot.com/2012/09/modelo-en-v.html>

Llamas, L. (2016). *Cómo Usar Un Acelerómetro En Nuestros Proyectos De Arduino*. Obtenido de Ingeniería, informática y diseño: <https://www.luisllamas.es/como-usar-un-acelerometro-arduino/>

Llamas, L. (2016). *Cómo usar un Giroscopio en nuestros proyectos de Arduino*. Obtenido de Ingeniería informática y diseño: <https://www.luisllamas.es/como-usar-un-giroscopio-arduino/>

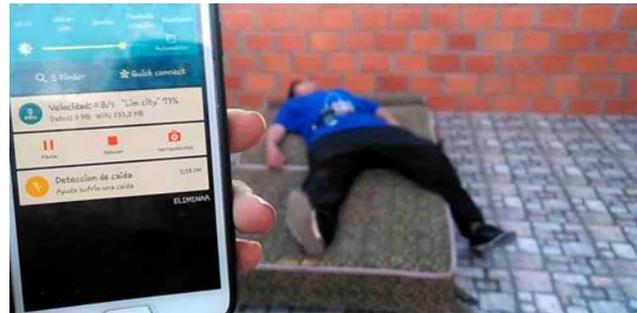
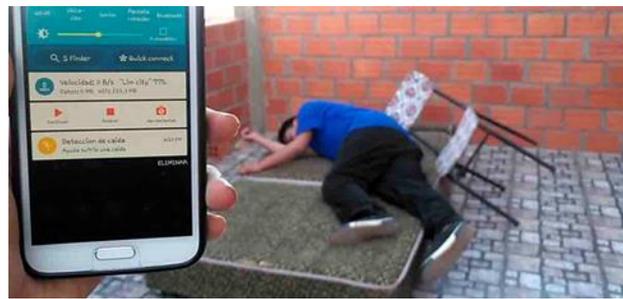
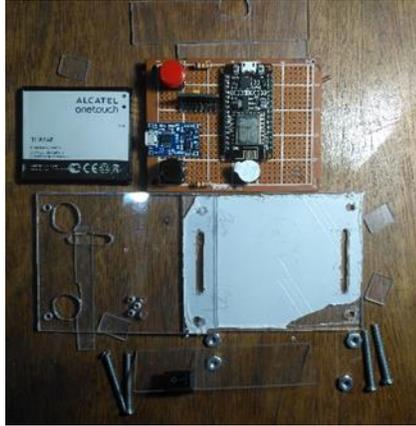
Llamas, L. (2018). *NodeMCU la popular placa de desarrollo con esp8266*. Obtenido de Ingeniería informática y diseño: <https://www.luisllamas.es/esp8266-nodemcu/>

- Llamas, L. (2018). *NodeMCU, la popular placa de desarrollo con esp8266*. Recuperado el 1 de febrero de 2020, de Ingeniería, informática y diseño: <https://www.luisllamas.es/esp8266-nodemcu/>
- López, S. (2020). *Firestore: qué es, para qué sirve, funcionalidades y ventajas*. Obtenido de Digital55: <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/>
- Mecafenix, F. (2018). *Sensor inercial o sensor imu*. Obtenido de Ingeniería Mecafenix: <https://www.ingmecafenix.com/automatizacion/sensores/sensor-inercial/>
- Medrano, S. C. (2014). *Detección de caídas para teléfonos inteligentes basados en algoritmos de detección de novedad*. Escuela Técnica Superior de Ingenieros Industriales. Obtenido de http://e-spacio.uned.es/fez/eserv/tesisuned:IngInd-Ctmedrano/MEDRANO_SANCHEZ_Carlos_Tomas_Tesis.pdf
- Mex, t. (2018). *El Modelo en V o de Cuatro Niveles*. Obtenido de Ingeniería de Software: http://ingenieriadesoftware.mex.tl/61885_modelo-v.html
- Mitchell, R. (2018). *Introducción al diseño de circuitos con Fritzing*. Obtenido de Maker Pro: <https://maker.pro/custom/tutorial/an-introduction-to-circuit-design-with-fritzing>
- Mora González, S. (2015). Entendiendo el Internet de las cosas. *TEC Instituto Tecnológico de Costa Rica*(23). Obtenido de https://revistas.tec.ac.cr/index.php/investiga_tec/article/download/2381/2169
- OMS. (2014). «Envejecer bien», una prioridad mundial. Obtenido de Organización Mundial de la Salud: <https://www.who.int/mediacentre/news/releases/2014/lancet-ageing-series/es/>
- Opinión . (2020). *800 mil bolivianos pasan la cuarentena solos, casi la mitad son adultos mayores*. Obtenido de Opinión diario de circulación nacional: <https://www.opinion.com.bo/articulo/pais/800-mil-bolivianos-pasan-cuarentena-solos-casi-mitad-son-adultos-mayores/20200405222202760484.html>
- Organización mundial de la salud . (2018). *Caídas*. Obtenido de Organización mundial de la salud : <https://www.who.int/es/news-room/fact-sheets/detail/falls>
- Padilla Roberto, Q.-R. V. (2015). Monitoreo y localización de personas extraviadas utilizando Arduino y GSM/GPS. (Artículo). Universidad Nacional Mayor de San Marcos, Lima, Peru.

- Peña , M. (2019). *Qué es el Internet de las Cosas y cómo afecta tu vida diaria*. Obtenido de Digital Trends: <https://es.digitaltrends.com/tendencias/que-es-el-internet-de-las-cosas/>
- Raigón , M. (2017). *Arduino, hardware y software para todos*. Obtenido de ticarte: <http://www.ticarte.com/contenido/arduino-hardware-y-software-para-todos>
- Rouse, M. (2017). *Internet de las cosas (IoT)*. Obtenido de TechTarget: <https://searchdatacenter.techtarget.com/es/definicion/Internet-de-las-cosas-IoT>
- Ruiz, M. (2017). *¿Qué es Firebase de Google?* Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-es-firebase-de-google/>
- Santiago, D. (2019). *Evolucion del Internet de las Cosas*. Obtenido de Telcel Empresa: <https://www.telcel.com//empresas/tendencias/notas/evolucion-del-internet-de-las-cosas>
- Shubham, P., & Samhita , G. (2016). *Uso del MPU9250 para obtener datos de movimiento en tiempo real*. Obtenido de Arduino project hub: <https://create.arduino.cc/projecthub/30503/using-the-mpu9250-to-get-real-time-motion-data-08f011>
- Tavizon, A., Guajardo, T., & Laines, C. (2016). *Iot, el internet de las cosas y la innovación de sus aplicaciones*. Obtenido de ResearchGate: https://www.researchgate.net/publication/326129401_IOT_el_internet_de_las_cosas_y_la_innovacion_de_sus_aplicaciones
- Tr4nsduc7or. (2014). *Tutorial de Arduino y MPU-6050*. Obtenido de Robologs: <https://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>
- Ucha, F. (2013). *Definición de Adulto mayor*. Obtenido de DefinicionABC: <https://www.definicionabc.com/social/adulto-mayor.php>
- Valois, M. (2018). *Qué es internet de las cosas y cómo funciona*. Obtenido de HostGator: <https://www.hostgator.mx/blog/internet-de-las-cosas/>
- Vázquez, Á. J., Martínez, S. W., Sánchez, L. R., & Sánchez, G. C. (2018). *Implementacion de un detector de caidas para su aplicacion en pacientes hospitalizados y personas de la tercera edad*. Universidad Autónoma del Carmen.

- Veloso, C. (2016). *Que es y para qué sirve Arduino*. Obtenido de Etools:
<https://www.electrontools.com/Home/WP/2016/04/20/que-es-y-para-que-sirve-arduino/>
- Volt, C. (2016). *Laser controlado por Acelerómetro ADXL345*. Obtenido de RogerBit:
<http://rogerbit.com/wprb/2016/03/robotica-laser-controlado-por-acelerometro-adxl345/>
- Zamora, J. (2016). *¿Qué es Firebase? La mejorada plataforma de desarrollo de Google*. Obtenido de El Androide Libre:
<https://elandroidelibre.elespanol.com/2016/05/firebase-plataforma-desarrollo-android-ios-web.html>

ANEXOS



DOCUMENTACIÓN