

Desarrollo de un Prototipo detector de caídas utilizando la placa Intel Galileo Generación I y el sensor MPU6050

Esteban Carnuccio¹, Waldo Valiente¹, Mariano Volker¹, Graciela De Luca¹,
Gerardo Garcia¹, Daniel Giulianelli¹, Sebastián Barillaro¹

¹Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina

{ecarnuccio, wvaliente, mvolker, gdeluca, ggarcia,
dgiunlian, sbarillaro }@unlam.edu.ar

Resumen: El objetivo de este estudio es exponer los avances conseguidos durante el desarrollo de un sistema prototipo que permite detectar la ocurrencia de una caída en una persona. El cual formará parte de un sistema integral orientado a Internet de las Cosas para controlar el estado de salud y vida diaria de las personas mayores de edad utilizando la placa Intel Galileo Generación I. Se describen los criterios tenidos en cuenta para elaborar el algoritmo de caídas, mencionando las fases por la que pasa el cuerpo en dicha situación. Además se describen qué tipos de actividades de la vida diaria han sido tenidos en cuenta para descartar falsos positivos, a través del análisis de mediciones, cómo se compone estructuralmente el prototipo y las dificultades encontradas durante su construcción, tanto de hardware como de software.

Palabras Clave: Acelerómetro, Giroscopio, detección de caída, Sensor MPU6050, Fases de caída.

1 Introducción

Las caídas no asistidas en forma oportuna en una persona anciana pueden provocar deterioro funcional, requiriendo de cuidadores o personas de apoyo. La prevención de caídas es tan importante que el ministerio de Desarrollo Social de la Nación a través la Dirección Nacional de Políticas para Adultos Mayores confeccionó una guía [1], donde el tema de caídas es parte principal de las recomendaciones antes situaciones de riesgo. Señalando que los efectos de las caídas pueden ser irreversibles sin una intervención adecuada y precoz. También la Organización Mundial de la Salud [2] posee un informe indicando que cada año un tercio de la población de mayores de 64 años sufre algún tipo de caída, siendo éstas unas de las más importantes causas de lesiones en adultos mayores y una de las principales causas de muerte por accidente en los mayores de 75 años. Una oportuna atención por parte de

servicios médicos o familiares, luego de una caída, puede brindar una recuperación y contención más rápida, mejorando la confianza, o hasta incluso salvándole la vida.

Desde la década de 1980 una parte de la investigación tecnológica se ha enfocado en crear una interfaz de humano a humano. Lo que dio lugar a la creación de la disciplina computación ubicua, cuyo objetivo es integrar la tecnología en el contexto de la vida cotidiana. Mediante Internet es posible que comunicar a los dispositivos individuales con cualquier otro dispositivo en el mundo y de esta forma diseñar sistemas de detección y aviso de caídas, que pueden ser clasificados en dos tipologías: los sensibles al contexto y los portátiles.

Los sistemas sensibles al contexto requieren de la instalación de sensores diseminados por la vivienda. Estas soluciones suele ser las más precisas ya que identifican el lugar de la caída y no necesita la persona cargar todo el día con un equipo (no intrusivo), pero es limitado por el alto costo de instalación y su rango de acción solo aplica en ámbitos cerrados. Los sistemas portátiles se basan en la incorporación de sensores en un aparato portable o vestible que supervisa las actividades de los adultos mayores en tiempo real, detectando caídas sobre la base de los cambios en algunos parámetros de movimiento. Los sensores utilizados para reconocer movimientos se componen de un acelerómetro y un giroscopio. Procesando la información sobre la orientación, la velocidad angular y la aceleración permitiendo describir el movimiento de una persona. Este es un enfoque es de bajo costo y fácil de emplear, no requiere de instalación, funciona tanto dentro como fuera de la casa. En contra partida al ser portable, puede ser olvidado o no activado por la persona mayor. Si es integrado a un celular puede tener reticencia a utilizarlo. Además un golpe involuntario en el sensor, o movimientos bruscos en determinadas condiciones pueden provocar una falsa alarma. Estas pueden ser reducidas mediante el uso combinado de sensores y consiguiendo con esto además una detección más precisa. Este tipo de dispositivo, permite además alcanzar el beneficio a una población de bajos recursos, ya que con un costo reducido puede comunicar al adulto mayor con un familiar o una persona cercana para conseguir pronta asistencia.

2 Desarrollo

2.1 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 [3] [4]. En la Figura 1 se pueden observar las cuatro 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. En la última etapa del algoritmo, se buscó un cálculo sencillo que convalide la caída real, desechando los falsos positivos, que puedan surgir.

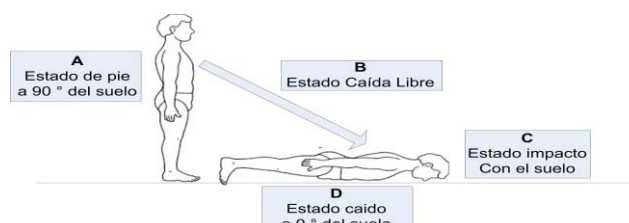


Fig. 1 Etapas de la Caída

Para poder calcular el ángulo que produce el movimiento del acelerómetro y el giroscopio dentro de un intervalo de tiempo, el método más empleado es el cálculo de los ángulos Pitch¹, Roll² y Yaw³, muy utilizado en aeronáutica. Oporto Díaz [3] calcula la posición de la persona al caerse utilizando fórmulas de seno y coseno para determinar la inclinación del individuo respecto a la vertical de los ejes X, Y y Z. Para ello en todo momento se ubica al sensor a la altura del pecho del individuo. En cambio Ramiro Blanco en “*El sistema de detección de caídas en un Geriátrico*” [4] determina si el sujeto ha sufrido un cambio de orientación, teniendo en cuenta qué eje se encuentra paralelo a la gravedad en el momento en que la persona finaliza su caída.

En esta investigación para determinar la inclinación que sufre la persona durante la caída, se comparan el ángulo inicial en que se encuentra la persona al ingresar en la fase de Reposo Inicial (A), con el que adquiere durante la fase de Reposo Final (D). El cálculo de los diferentes ángulos se realiza aplicando producto escalar de los vectores de aceleración, correspondientes al estado (A) y con el vector de aceleración que produce el movimiento en un instante determinado. Para ello se utiliza la fórmula (1).

$$\cos \alpha = \frac{(Acelx_{t0})^2 \cdot (Acelx_{tn})^2 + (Acely_{t0})^2 \cdot (Acely_{tn})^2 + (Acelz_{t0})^2 \cdot (Acelz_{tn})^2}{\sqrt{(Acelx_{t0})^2 + (Acely_{t0})^2 + (Acelz_{t0})^2} \cdot \sqrt{(Acelx_{tn})^2 + (Acely_{tn})^2 + (Acelz_{tn})^2}} \quad (1)$$

El resultado del ángulo calculado con la fórmula (1), es posteriormente verificado con el valor obtenido del Pitch y Roll. El Yaw no se está utilizando debido a que para saber si una persona se cayó, no es necesario conocer el ángulo que obtiene la persona entorno al eje Z.

¹ Angulo de rotación en eje Y

² Angulo de rotación en eje X

³ Angulo de rotación en eje Z

2.2 Prototipo

El prototipo realizado en el proyecto consta de una placa Intel Galileo Gen 1, con sistema operativo Linux Yoctos X.X.

Para detectar caídas se utiliza un acelerómetro con giroscopio MPU6050 y para lograr la autonomía se utiliza una batería especial de 8v Lithium, dado que la placa Galileo necesita sólo 5 volts de alimentación, se construye un circuito reductor con transistor LM2576. Además se le adicionó un botón pulsador como reset para poder reiniciar el sistema operativo en casos de necesidad. También consta de una conexión por wifi para organizar las mediciones en tiempo real. El diagrama de conexión se muestra en la **Figura 2**.

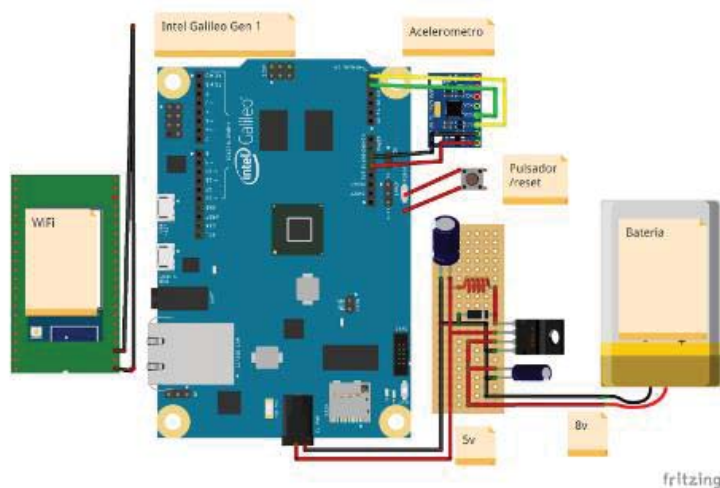


Fig. 2 Diagrama de conexión del prototipo

Uno de los problemas que surgió al utilizar el prototipo, se debió a la ubicación del acelerómetro dentro del gabinete del prototipo, provocó errores por descalibración del sensor MPU6050. Debido a que es sensible a los cambios bruscos de temperatura (rango de trabajo, -40°C a 85°C) [5]. Su hoja de datos especifica que el desvío de las mediciones del acelerómetro es de $\pm 0.02\%$ por grado centígrado y el giroscopio de $\pm 20\%$ por grados/segundos. El que afecta indirectamente a la calibración del sensor. Por ese motivo, para que el MPU6050 no se vea afectado a la temperatura que genera el microprocesador, se aisló el sensor y se lo colocó en la parte exterior del prototipo. Debido a que el microprocesador en un uso intensivo de la CPU llega a alcanzar una temperatura de 70°C . De esta forma gracias a la aislación, se logró que se afecten mínimamente las mediciones, realizadas en distintas temperaturas ambiente una vez calibrado el sensor. Además como el rango de sensibilidad de los sensores fue aumentado al máximo soportado por estos, 16G y $2000^{\circ}/\text{seg.}$, se concluyó que las mediciones no serían afectadas por la temperatura.

2.3 Ubicación del prototipo.

En la primera etapa las mediciones fueron realizadas [6] sin tener en cuenta un eje de referencia, con lo cual resultó dificultoso poder determinar correctamente el ángulo inicial y final en que se encontraba la persona en determinado momento. Esto se debió a que el prototipo no se encontraba en un posición fija durante las mediciones. Luego de diferentes pruebas se resolvió ubicarlo sujeto en la cintura del individuo **Figura 3**, resolviendo este inconveniente ya que de esta manera el eje Y sería siempre el afectado por la gravedad en lugar del eje Z. Luego de esto se realizó el muestreo de mediciones de las caídas de una persona, para poder obtener así los valores correctos de aceleración y orientación.



Fig. 3 Ubicación del prototipo en una persona

2.4 Mediciones realizadas

Una persona puede realizar distintas actividades diarias, cada una de ellas involucra distintos movimientos. A lo largo de dicha acción la persona va adquiriendo diferentes valores de aceleración y de orientación, que deben ser tenidos en cuenta al momento de analizar si se produjo o no una caída. Permitiendo de esta manera evitar falsos positivos durante la ejecución del algoritmo de detección. Por eso dos individuos diferentes contextura física han realizado distintas actividades, entre ellas caídas, y se han capturado los valores devueltos por el sensor MPU6050, con la finalidad de poder determinar correctamente el funcionamiento del algoritmo de detección. Los sujetos de prueba son de sexo masculino, de aproximadamente 35 años y de diferentes alturas. El primero mide 1.80 metros de altura y pesa 80 kg, mientras que el segundo mide 1.70 metros y pesa 72 kg. Las actividades realizadas por los sujetos de pruebas fueron las siguientes:

- **Actividades de la vida diaria:** Caminar, saltar, sentarse y levantarse de una silla, subir y bajar por una escalera
- **Actividades de caídas:** Caerse estando de pie hacia adelante, atrás o de costado. También se realizaron mediciones cayéndose hacia adelante o de costado estando sentado .

A través de las mediciones de las actividades realizadas, se pueden determinar los patrones que siguen las personas al realizar dichas tareas y observar cómo se comporta el cuerpo humano cuando sufre una caída. Por ese motivo a continuación se muestran los tres gráficos más significativos, obtenidos con los valores de las mediciones realizadas. Por un lado se muestran gráficamente los valores resultantes del cálculo de la norma de la aceleración [6]. Mientras que por otro, se muestra gráficamente el ángulo que tiene el cuerpo del sujeto de prueba en determinado momento.

a) Al caminar.

En la **figura 4** se muestra la norma de la aceleración que obtiene una persona al caminar. Como se puede observar la aceleración nunca supera los 2.5G y no llega a disminuir a los 0G. Este detalle es importante ya que nos sirve para poder determinar el alcance del algoritmo de detección.



Fig. 4 Norma de la aceleración al caminar.

b) Una persona de pie , que cae hacia adelante.

En la **figura 5** se muestran dos gráficos, el primero expone los valores de la norma de la aceleración cuando una persona se encuentra de pie y se cae hacia adelante. Mientras que el segundo gráfico muestra las variaciones de ángulos que va adquiriendo el cuerpo de la persona a medida que transcurre la caída.

Cómo se puede observar en **A** la persona se encuentra en el estado de Reposo Inicial. En esta fase, se encuentra quieta, sin embargo su aceleración es de 1G debido a que se ve afectado su eje Y por la fuerza de gravedad. Además se puede visualizar en **A**, que el ángulo inicial es de 0° . Luego, una vez que la persona pierde el equilibrio, empieza en **B** la fase de Caída Libre. En la cual se observa una disminución brusca de la aceleración, desde 1G hasta casi llegar a 0G. A continuación, apenas la persona toca el suelo se produce la transición al estado de impacto **C**, anteriormente mencionada. En esta fase se puede observar un pico abrupto en la aceleración de casi 7G, superando de esta forma el límite de los 2.5G que se puede llegar a alcanzar cuando se camina. Finalmente la aceleración decrece,

normalizándose hasta 1G. Produciéndose así el cambio de estado a la etapa Reposo Final **D**. Esta última etapa se diferencia de la fase **A**, debido a la existencia de un cambio de orientación en el cuerpo de la persona, de 0° a 90° . Indicando así que efectivamente la persona se ha caído al suelo.

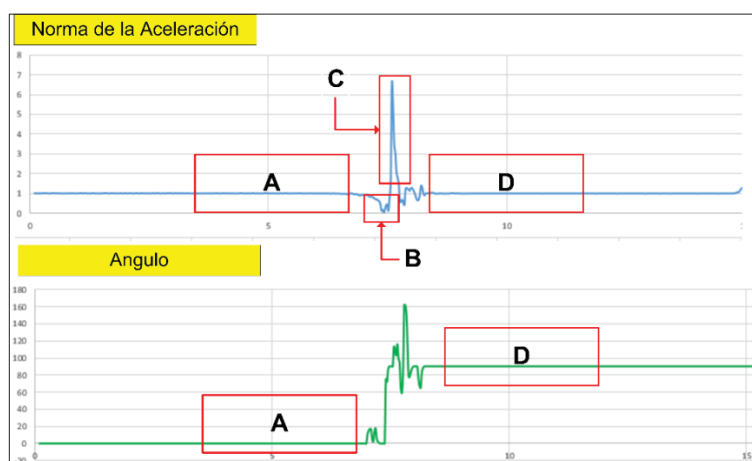


Fig. 5 Norma de la aceleración y ángulo de la persona al caerse hacia adelante.

2.5 Solución de software

Al comienzo se desarrolló el software embebido en el prototipo utilizando el lenguaje de programación Wiring de Arduino. Sin embargo por cuestiones de desempeño al utilizar el prototipo con la placa Intel Galileo Gen I, se decidió cambiar este lenguaje por Node.js, ya que esta plataforma emula determinados componentes de Arduino. No obstante en determinadas situaciones se utilizan sketches⁴ de Arduino para realizar tareas muy específicas. Desde Node.js se está utilizando la biblioteca [7] de UPM para leer el sensor MPU6050 en la placa Intel Galileo.

Para el desarrollo del detector de caídas utilizando Node.js, se está utilizando programación basada en eventos, debido a que este lenguaje se apoya en este paradigma. En consecuencia, el algoritmo de detección de caídas se basa en los eventos mencionado anteriormente. Por ese motivo se muestra en la **figura 6** la transición de estados y eventos que deben cumplirse en nuestro algoritmo para que sea detectada la caída de una persona. Se puede observar además, que si se da una secuencia completa, desde el estado reposo inicial hasta el estado reposo final, significa que se ha identificado una caída de la persona observada.

En las mediciones obtenidas primeramente, se había configurado al acelerómetro del MPU6050 con un grado de alcance de 2G y al giroscopio con $250^\circ/\text{seg}$, con el objeto de poder tener mayor sensibilidad al tomar las mediciones. Sin embargo,

⁴ **Sketch** - Parte de código fuente listo para abrir y ser cargado sobre el dispositivo, con el entorno de desarrollo integrado de Arduino.

durante el desarrollo del prototipo se observó que dicho rango no era suficiente, debido a que a una caída normal no superaba los 2G de aceleración y una persona caminando normalmente alcanzaba fácilmente ese valor. Por dicho motivo fue necesario cambiar el rango soportado por el acelerómetro a 16G y a 2000°/seg el giroscopio. De esta forma se intenta mejorar el rango de detección del MPU6050, descartando a su vez aceleraciones mínimas que se obtienen al mover levemente el prototipo.

El MPU6050, cómo la mayoría de los sensores, presenta desviaciones en los valores que vaya sensando a lo largo del tiempo. Pueden ser diversas las causas de estas desviaciones, también conocidas como offsets, ya que su origen puede deberse a las soldaduras que presenta el integrado desde su momento de fabricación, o a cambios de temperatura y humedad, entre otras. Por lo tanto resulta indispensable su calibración antes de la comenzar a utilizarlo, siendo necesario realizar este paso, para encontrar los offsets correctos para calibrar el sensor.

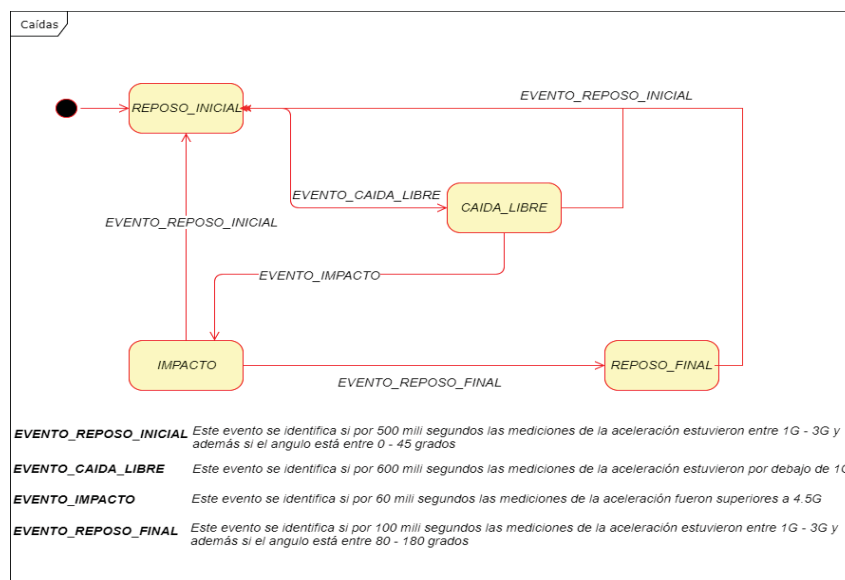


Fig. 6 Secuencia de transición de estados para detectar una caída

En primera instancia de la calibración del MPU6050 se había conseguido realizar utilizando el sketch calibrador desarrollado por Naylampmechatronics [8]. Sin embargo, durante el desarrollo del módulo detector de caídas, se descubrió que este programa era poco eficiente, debido a la demora excesiva en su ejecución. Esta situación es poco performante y de poca usabilidad si el prototipo es utilizado por un usuario adulto. Por ese motivo actualmente se reemplazó este mecanismo de calibración utilizando el sketch desarrollado por Luis Ródenas [9]. El cual permite obtener los offsets correctos para cada eje y almacenarlos en los registros offsets del procesador DMP del MPU6050 [5]. Para poder ejecutar este código, al igual que cualquier otro calibrador de acelerómetro y giroscopio, es necesario que la primera

vez que se utiliza el prototipo se lo coloque sobre un plano horizontal durante su ejecución, apuntando el eje Z positivo hacia el techo. Después de que sean almacenados los offsets en el DMP, cada vez que se ejecute el programa de Node.js, se utilizará indirectamente los offsets que se encuentran en el DMP restándolos automáticamente a los valores leídos del acelerómetro y del giroscopio. De esta manera se están obteniendo las mediciones sin desviación. En el programa de Node.js se debió emplear la biblioteca [7] para poder utilizar el MPU6050 desde ese lenguaje de programación.

Es importante mencionar, que el sketch de Luis Ródenas está desarrollado para funcionar con una escala de 2G. Por lo tanto al modificar la sensibilidad del acelerómetro a 16G, fue necesario modificar este sketch para que el valor de los offsets sea calculado de acuerdo a la escala con la que se está trabajando.

2.6 Desempeño del funcionamiento de Node.js en un sistema embebido.

Uno de los problemas encontrados durante el análisis de las primeras mediciones realizadas [6], fue el intervalo de tiempo en el que se habían capturado los valores sensados por acelerómetro y giroscopio, debido a que se realizaba repetidamente una lectura del sensor cada 50 milisegundos. Por ese motivo la cantidad de datos capturados, en el intervalo de tiempo, no eran suficientes para poder determinar si se había efectuado una caída. Con lo cual fue necesario bajar el tiempo de muestreo a 20 milisegundos. Siendo necesario realizar adaptaciones en el código fuente para subsanar las limitaciones que presenta Node.js. Una de las modificaciones necesarias en el programa fue paralelizar la lectura del sensor MPU6050 que se realizan desde Node.js. Dado que este lenguaje se ejecuta en un único hilo de ejecución, se debió utilizar la biblioteca *child_process* para poder paralelizar su ejecución utilizando procesos [9]. No obstante, se pudo observar que cualquier programa realizado en Node.js que lee valores de sensores y se ejecute en la Placa Intel Galileo Gen I, es poco performante en cuanto a rendimiento. Debido a que consume considerables recursos de hardware, como CPU y memoria. Pero se consigue disminuir mínimamente su consumo, si se lo paraleliza con procesos hijos. Esto es debido a que Node.js no fue desarrollado para ser utilizado en sistemas embebidos, sino principalmente para el desarrollo principal de servidores web. Sin embargo debido a que el prototipo del detector de caídas se está desarrollando en una placa Intel Galileo, que posee suficiente recursos de hardware para poder ejecutar el programa realizado en Node.js sin percance, se decidió que era conveniente seguir desarrollando el módulo detector de caídas utilizando este lenguaje de programación.

3 Conclusiones

De acuerdo a lo expuesto se ha podido determinar los patrones de movimientos que tiene una persona al realizar sus actividades diarias y los diferentes tipos de caídas que puede sufrir ante determinados sucesos. A partir del análisis realizado se pudieron detectar los eventos que permite diferenciar las caídas de una actividad rutinaria,

diseñando un algoritmo de detección de caídas con las restricciones antes mencionadas, que puede descartar falsos positivos teniendo en cuenta las fases de la caída.

Se continuará en la investigación con la implementación del algoritmo de detección, realizando pruebas unitarias con el prototipo construido, teniendo en cuenta las distintas acciones que puede realizar el adulto en su vida diaria, y ante la detección de una caída notificarle automáticamente al familiar dicha ocurrencia a través de dispositivos móviles. Además se pretende agregar mayor precisión incorporando otros tipos de sensores al prototipo en el futuro, como pulso cardíaco que se incorporará a nuestro algoritmo de caída. En fases siguientes de la investigación, a partir del prototipo, se proyecta elaborar un producto comercial útil para la sociedad, utilizando componentes de hardware más específicos, en vez de placas de desarrollo que son empleadas para prototipado. Se espera que éste proyecto realice un aporte considerable a una necesidad concreta de nuestra sociedad, la supervisión de personas ancianas, utilizando metodologías IoT y poniéndolas a disposición de los individuos que no dispongan de recursos o tiempo para ésta actividad, teniendo la premisa de que la solución utilice herramientas y/o dispositivos masivos para reducir costos en la solución final, permitiendo a personas de cualquier nivel socioeconómico la posibilidad de acceder a estos dispositivos.

4 Referencias

- [1] M.L.Roqué, Guía para el autocuidado de la salud de las personas mayores, Buenos Aires: Secretaría Nacion de la Niñez, 2010.
- [2] Casiliari y Oviedo-Jiménez, Automatic Fall Detection System Based on the Combined Use of Smartphone and a Smartwatch, Malaga, España: Universidad de Málaga, 2015.
- [3] E. Oporto Díaz, «Diseño de un sistema inalámbrico de detección de caídas aplicado a personas de la tercera edad basado en acelerómetro y teléfono móvil,» Lima-Peru, 214.
- [4] R. Blanco, «Sistema de detección de caída en personas de la tercera edad para uso en centro geriaticos,» Bogota, 2010.
- [5] I. InvenSense, *MPU-6000 and MPU 6050 Product Specification Revision 3.4.*, Sunnyvale: Inc.InvenSense, 2013.
- [6] S. Barillaro, W. Valiente, G. De Luca, E. Carnuccio, G. García, M. Volker, D. Guilianelli y N. Casas, «Diseño de sistema IoT de monitoreo para personas mayores,» de *XIX Workshop de Investigadores en Ciencias de la Computación(WICC 2017)*, Buenos Aires, 2017.
- [7] Intel, «http://iotdk.intel.com/docs/master/upm/group__mpu9150.html,» [En línea].
- [8] S. Barillaro, W. Valiente, G. De Luca, E. Carnuccio, G. Gracia, M. Volker y D. Guilianelli, «IoT fall detection system for the elderly using,» de *IEEE CACIDI 2016 - IEEE Conference on Computer Sciences*, IEEE, 2016.
- [9] L. Ródenas, «<https://www.i2cdevlib.com/forums/topic/96-arduino-sketch-to-auto>