

# Comparación de Dos Algoritmos de Detección de Caídas Utilizando Teléfonos Inteligentes

## ***Lynette Drusilla Layne***

Universidad Autónoma de San Luis Potosí, Av. Salvador Nava Mtz s/n, Zona Universitaria,  
San Luis Potosí, (444)826-2486  
*rmariela@fciencias.uaslp.mx*

## ***Ruth M. Aguilar-Ponce***

Universidad Autónoma de San Luis Potosí, Av. Salvador Nava Mtz s/n, Zona Universitaria,  
San Luis Potosí, (444)826-2486

## ***J. Luis Tecpanecatl-Xihuitl***

Universidad Autónoma de San Luis Potosí, Av. Salvador Nava Mtz s/n, Zona Universitaria,  
San Luis Potosí, (444)826-2486

## ***Hector Raúl Moncada González***

Universidad Autónoma de San Luis Potosí, Av. Salvador Nava Mtz s/n, Zona Universitaria,  
San Luis Potosí, (444)826-2486

## **Resumen**

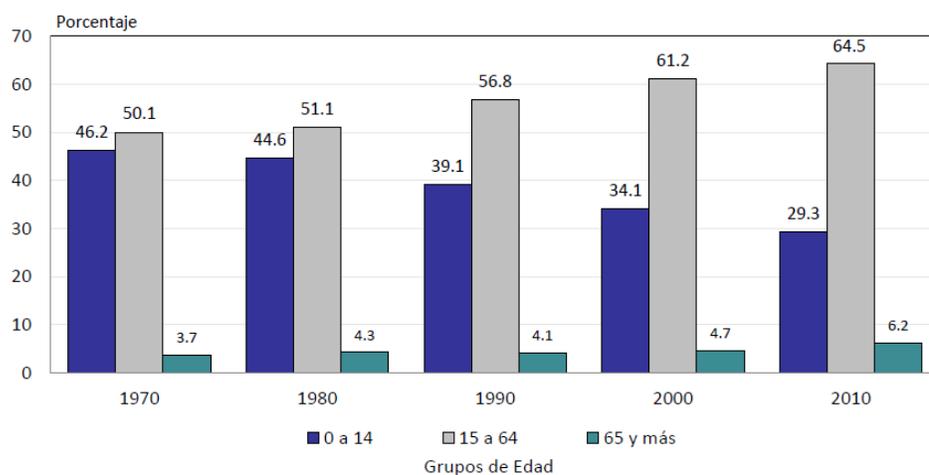
Aproximadamente un 30 % de las personas de la tercera edad suelen sufrir una caída importante al año, resultando en un alto índice de hospitalización y una alta tasa de mortalidad. Esta situación presenta un reto importante en el marco de salud pública, por lo que se requiere un sistema eficaz de detección de caídas para proporcionar un apoyo urgente y reducir significativamente los costos de atención médica asociados a las caídas. Considerando esta problemática, se presenta el estudio de la potencia utilizada por dos Algoritmos de Detección de caída. El primer sistema detecta las caídas mediante la diferenciación de las señales de acelerómetro; mientras el segundo algoritmo aísla la frecuencia que indica que se ha presentado una caída mediante una serie de filtros. Para

la implementación de los algoritmos, se utilizaron teléfonos inteligentes que cuentan con acelerómetros triaxiales. Ambos algoritmos se implementaron en el lenguaje Java dentro del marco de trabajo de Android. Se concluyó que el primer algoritmo es más eficiente en términos del consumo de energía debido a que en las pruebas realizadas consumió 2.6% menos potencia que el segundo algoritmo.

**Palabras Claves:** Consumo de Potencia, Detección de Caídas, Diferenciación de Señales, Procesamiento Digital de Señales, Sistemas Embebidos.

## 1. Introducción

La población de México ha ido envejeciendo y esto se ve reflejado en el aumento de la población de 65 años o más que representa actualmente el 6.2% del total, según datos reportados por la Secretaría de Gobernación de los Estados Unidos Mexicanos y el Banco Mundial [1]. Se considerará la población de la tercera edad a todas las personas de 65 años en adelante. En cuarenta años este grupo de población casi duplicó su participación relativa, mientras que la población infantil disminuyó en casi 17 puntos porcentuales como se refleja en la Fig. 1. [2]



**Fig. 1. Distribución porcentual de la población por grupos de edad, (1970-2010).**

Las personas de la tercera edad suelen sufrir al menos una caída al año según reporta la Organización Mundial de la Salud. El 10 a 25% de las caídas en el adulto mayor resultan en fracturas, mientras que el 5% requiere hospitalización. La mayoría de las caídas ocurre en casa lo cual representa aproximadamente un 62% del total de caídas; mientras que el 26% de estas ocurren en la vía pública. El 25% de los adultos mayores que sufren fractura de cadera mueren en los siguientes seis meses [3].

Considerando los resultados estadísticos del crecimiento de la población de la tercera edad, se requiere de un método de atención inmediata en caso de que exista una caída. Dado que la mayor cantidad de caídas ocurren en casa, se requiere de un método de atención inmediata. Tomando en cuenta este escenario, se han propuesto varias aproximaciones a la solución de este problema, las cuales pueden ser clasificadas en tres categorías: Sistemas basados en visión computacional, dispositivos corporales (wearable devices) y dispositivos ambientales [4].

Las cámaras se han incluido en varios sistemas de asistencia en casa debido a la gran cantidad de información que pueden proporcionar y su costo se ha decrementado en los últimos años. Adicionalmente, las cámaras son menos intrusivas, detectan múltiples eventos simultáneamente y el video almacenado se puede usar para asistencia remota y verificación posterior al evento. Las propuestas en esta rama incluyen, detección del movimiento de la cabeza, deformación de la persona bajo supervisión y detección de inactividad. La detección de la inactividad usa cámaras estereoscópicas para obtener la ubicación y orientación de la persona. Si estas variables no cambian en un periodo determinado, se declara como un cuerpo inactivo y una alarma es activada [5]. Los algoritmos de análisis del cambio de forma identifican cuando una persona está de pie, sentada o acostada. En caso, de que se encuentre acostada por un periodo extendido de tiempo, se considera que ha ocurrido una caída. Finalmente, la detección del movimiento de la cabeza utiliza un método de seguimiento de la cabeza a través de un filtro de

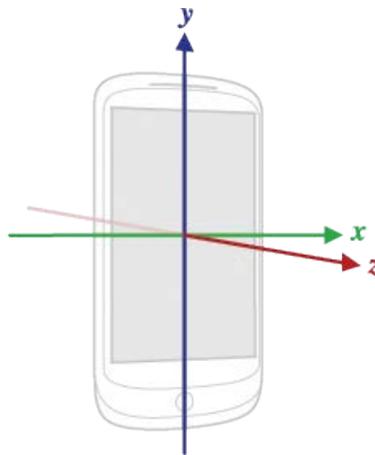
partículas. Se calcula la velocidad horizontal y vertical de la cabeza. En una caída se asume que el movimiento vertical es más rápido que el horizontal, por lo que se establecen un umbral en las velocidades calculadas para determinar cuándo se presenta una caída. Sin embargo, los sistemas basados en cámaras requieren que se establezca una red de cámaras dentro de la casa, lo cual implica un costo considerable. Adicionalmente, solo se pueden detectar caídas dentro del espacio de visión de las cámaras, lo cual implica que las caídas que sucedan fuera de este espacio no serán detectadas.

Los dispositivos ambientales contienen sensores y unidades de procesamiento de señales para monitorear algún evento en específico. Se han propuesto varias aproximaciones para la detección de caídas incluyendo sensores de presión, sensores infrarrojos y sensores de vibración. Los sensores infrarrojos son colocados en un arreglo que permita detectar los objetos en movimiento. Estos sensores evitan calcular la detección del fondo estático, dado que solo detectan objetos cálidos. Por medio del arreglo de sensores se determina la localización, tamaño y velocidad del objeto en movimiento. Estos datos son analizados y se determina cuando la persona bajo supervisión ha caído. Los sensores de presión se han colocado en los colchones para determinar cuando una persona se encuentra presente en la cama o está en riesgo de caer. Los dispositivos ambientales sufren del mismo problema que los sistemas basados en visión, solo pueden detectar caídas dentro del espacio de monitoreo. El costo de estos sistemas es alto y requiere de acondicionar el espacio vital de las personas bajo supervisión.

Finalmente, los dispositivos corporales o vestibles (wearable devices) son dispositivos embebidos en la ropa del usuario. Los teléfonos inteligentes y tabletas han dominado el mercado de dispositivos móviles en los últimos años. Sin embargo, estos dispositivos serán complementados a través de dispositivos corporales que puedan proveer de información adicional a los usuarios, tales como ritmo cardiaco, ayuda auditiva o monitores de actividad. Adicionalmente, estos dispositivos pueden enviar información adicional al usuario, tales como los google glasses, los cuales aumentan la realidad por

medio de introducir análisis de imágenes para proveer información de lo que el usuario está observando. Dentro de los trabajos propuestos se ha desarrollado un bastón que puede detectar las caídas y medir la velocidad de paso mediante un giroscopio embebido en el bastón. Uno de los sensores que se han empleado más en la detección de caídas es el acelerómetro. El acelerómetro de tres ejes mide la fuerza de aceleración experimentada en el eje vertical, horizontal y profundidad. El acelerómetro se ha localizado en la cintura, cabeza o muslo de la persona bajo supervisión y se han realizado varios algoritmos para determinar la caída. Los dispositivos corporales nos permiten detectar la caída independientemente del lugar donde se encuentre la persona bajo supervisión. El costo del sistema es más bajo debido a que solo se requiere menos sensores dado que solo se necesita monitorear a la persona y no una amplia zona de movimiento de la persona como sucede en las propuestas anteriores.

Nuestra propuesta consiste en detectar la caída a través de analizar las señales de acelerómetro obtenidas mediante un teléfono inteligente. Esta aproximación corresponde a los dispositivos corporales. No se desarrolló un sistema adicional dado que los teléfonos inteligentes cuentan con acelerómetros triaxiales, así como unidad de procesamiento y comunicación.



**Fig. 2. Sistema de Coordenadas para el Dispositivo.**

En las siguientes secciones se describe la implementación de los algoritmos de detección de caídas seleccionados en Android y su análisis de potencia consumida. Mientras que la sección 3 presenta los resultados alcanzados y las conclusiones son presentadas en la sección 4.

## **2. Implementación de los Algoritmos de Detección de Caídas**

Los teléfonos inteligentes son una plataforma ideal para la implementación de algoritmos de detección de caídas dado que contienen sensores, unidad de procesamiento y comunicación. Estos dispositivos tienen un ambiente maduro tanto en hardware como software. Adicionalmente, no se solicita que los usuarios tengan que incorporar dispositivos adicionales y más aún estos dispositivos son de uso común.

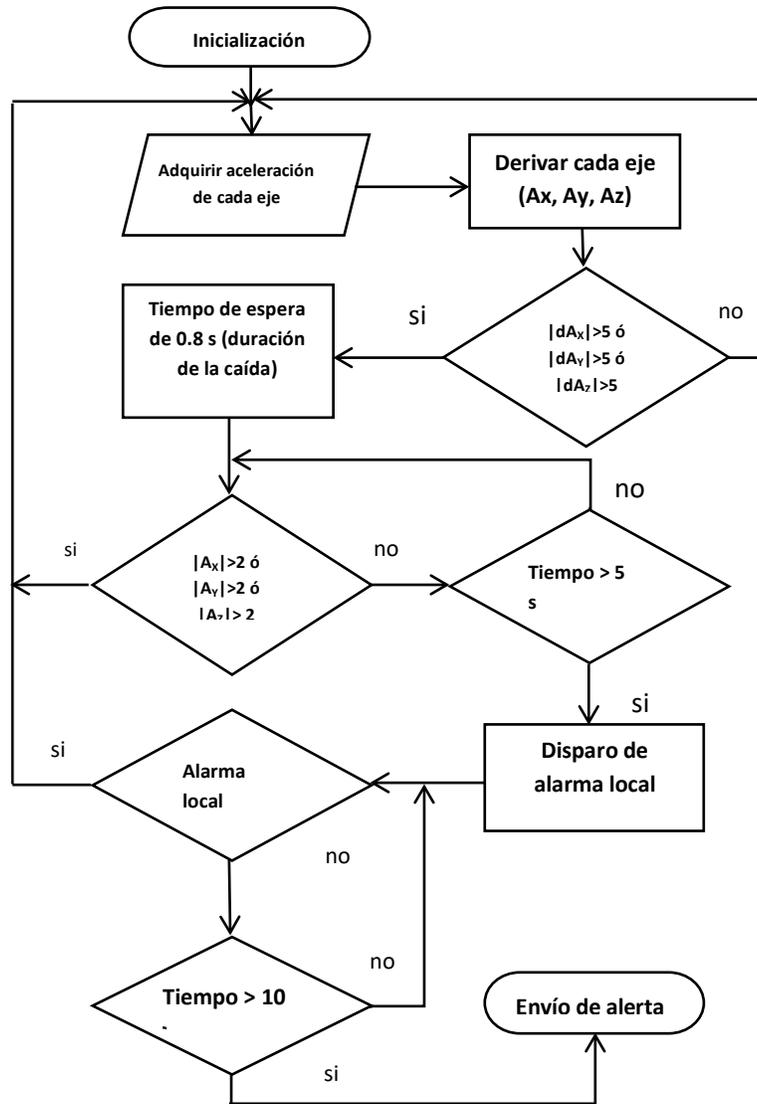
El acelerómetro mide la aceleración aplicada al dispositivo y se encuentra presente en todos los teléfonos inteligentes. Los acelerómetros triaxiales miden la aceleración en las tres direcciones como se muestra en la Fig. 2.

### **2.1 Detección de Caídas Mediante Diferenciación**

Una de las características fundamentales de la señal generada durante una caída, consiste en un cambio brusco de la aceleración ( $A$ ) en uno o más ejes. Esta particularidad puede ser explotada en el algoritmo de detección, utilizando únicamente la diferenciación en cada uno de los ejes, operación muy sencilla que requiere de recursos mínimos del sistema. La Fig. 3 ilustra el diagrama de flujo de este algoritmo. El procedimiento es obtener el valor absoluto de la derivada en cada una de las direcciones  $x$ ,  $y$  y  $z$ . Se determinó experimentalmente que un cambio equivalente a  $5g$  ( $g = 9.81 \text{ m/s}^2$ ) puede indicar una caída, lo cual es expresado en la siguiente ecuación:

$$Caida = \begin{cases} 1 & \text{Si } \left| \frac{dA(x)}{dx} \right| \geq 5 \text{ o } \left| \frac{dA(y)}{dy} \right| \geq 5 \text{ o } \left| \frac{dA(z)}{dz} \right| \geq 5 \\ 0 & \text{De lo contrario} \end{cases} \quad (1)$$

En caso de que el valor absoluto de la razón de cambio de la señal de acelerómetro en alguno de sus ejes sobrepase los 5g, se considera que ha ocurrido una caída. Sin embargo, no se lanzará una alarma a menos que después de registrado el evento no hay cambios en la señal de acelerómetro. La señal se monitorea después de 0.8 segundos a partir de la detección de caída. Esto se debe a que la caída toma alrededor de 1 segundo en realizarse.



**Fig. 3. Diagrama de Flujo del Algoritmo para la Detección de Caída mediante diferenciación.**

La segunda etapa del algoritmo es comprobar si la magnitud de la aceleración en cualquiera de los ejes es mayor a 2g, de ser así se asume que hay actividad normal tras la caída o tras una actividad que aparentó serlo; de lo contrario, se infiere que la persona no ha podido incorporarse y se disparará un alarma local. En este momento el teléfono inteligente sonará por un tiempo predeterminado por el usuario al momento de la

configuración de la aplicación. Si la persona se ha recuperado de la caída, presionará un botón que desactivará la alarma de lo contrario al concluir el tiempo establecido la aplicación enviará un mensaje de texto a un número telefónico previamente capturado.

## 2.2 Detección de Caídas Mediante Filtrado

Se realizó la caracterización de la caída por medio del análisis en frecuencia de la señal. Para lograr esto, se simularon varias caídas con sujetos sanos de entre 20 a 30 años. Se obtuvieron las señales de acelerómetro utilizando el Ez430 Chronos Watch de Texas Instruments [10]. Se tomó la aceleración total que percibe el sensor, la cual es calculada mediante la siguiente ecuación:

$$A_T(n) = \sqrt{|A_x(n)|^2 + |A_y(n)|^2 + |A_z(n)|^2}$$

Donde  $A_x(n)$ ,  $A_y(n)$ , y  $A_z(n)$  son la señales de aceleración en los ejes  $x$ ,  $y$  y  $z$  en el tiempo  $n$ . En base a este análisis se determinó que las caídas se localizan por encima de los 60Hz. Por lo que se utilizó un filtro pasa-altas con frecuencia de corte en 60Hz. Una vez separada la componente en frecuencia que puede indicar que existe una caída, se determinó la duración promedio de una caída. Se utilizó un filtro promediador para eliminar las variaciones en la señal de un ancho menor a número de muestras correspondientes a la duración de la caída cuya función de transferencia se define en la ecuación 2. Ese número de muestras se determinó experimentalmente y se estableció en 8 muestras.

$$H_{pr}(z) = \frac{1}{8} \sum_{i=0}^7 z^{-i} \quad (2)$$

Finalmente se detecta una caída a través de un umbral pre-establecido por medio de los experimentos realizados. La Fig. 4 ilustra el procedimiento realizado.



**Fig. 4. Diagrama de los bloques utilizados.**

Una vez detectada la caída, se monitorea la magnitud de la señal de acelerómetro. Si esta magnitud no percibe cambios por arriba de 2g, entonces se activa una alarma. Al igual que en el algoritmo anterior, se manda un mensaje de texto solo si el usuario no puede desactivar la alarma.

### 2.3 Calculo de la Potencia Consumida

La duración de la batería en los teléfonos inteligentes es primordial para que el dispositivo pueda ser utilizado por un mínimo de 8 horas sin requerir recargarlo. Para garantizar su utilización es necesario que se determine el consumo energético de las aplicaciones que se ejecutan en el dispositivo. Dado que la detección de caídas será una actividad que se estará ejecutando todo el tiempo en el dispositivo es necesario determinar el consumo de potencia requerido. PowerTutor es una herramienta diseñada para ofrecer una manera de calcular la eficiencia energética del software para teléfonos inteligentes, la interfaz de la aplicación se muestra en la Fig. 5 [9]. La aplicación permite a los desarrolladores de software ver el impacto de los cambios de diseño en la eficiencia de energía [8].

PowerTutor utiliza un modelo de consumo de energía construido por mediciones directas en el control cuidadoso de los estados de administración de energía del dispositivo. El modelo de potencia de PowerTutor fue construido en los teléfonos HTC G1, HTC G2 y Nexus uno. Se puede ejecutar en otros teléfonos, pero las estimaciones de consumo de

energía serán aproximadas. Se utilizó esta aplicación para determinar el consumo de potencia de los dos algoritmos implementados.

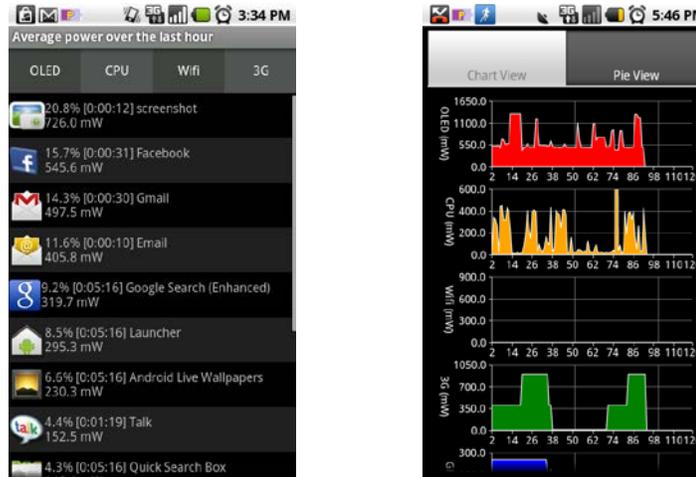


Fig. 5. Interfaz de PowerTutor.

### 3. Resultados

La implementación de ambos algoritmos sobre el sistema operativo Android se realizó con base a los diagramas de flujo mostrados en las Fig. 3 y 4 respectivamente. La implementación se realizó en código de java para crear dos aplicaciones para la plataforma Android 2.3.3 con API 10. Las pruebas se realizaron en dos teléfonos inteligentes: un Samsung Galaxy S1 y un Sony Xperia S. Para diferenciar los resultados obtenidos por cada algoritmo se nombraron de la siguiente manera, el primer algoritmo de detección mediante diferenciación, el nombre de la aplicación es Sensorwalk y el segundo algoritmo de detección mediante filtros, el nombre de la aplicación es DetectFall2.

El teléfono inteligente muestrea los datos del acelerómetro a una velocidad mayor a la velocidad de procesamiento de los datos. Por lo cual, se requiere de una implementación

multi-hilo bajo el esquema del productor-consumidor. Este tipo de implementaciones nos permite que los datos del acelerómetro se produzcan a la misma velocidad, garantizando una frecuencia de muestreo uniforme. Mientras que el segundo hilo puede procesar los datos a su propia velocidad. Sin embargo, este escenario requiere de sincronizar ambos hilos. El productor y consumidor comparten un buffer de tamaño finito. La tarea del productor es generar datos y almacenarlos; mientras que el consumidor toma datos uno a uno. El problema consiste en que el productor no añada más productos que la capacidad del buffer y que el consumidor no intente tomar un producto si el buffer está vacío.

Se realizaron una serie de experimentos para determinar la efectividad de detección de los algoritmos utilizados y su consumo energético. Participaron 4 personas sanas simulando caídas en un ambiente controlado. Cada persona realizó una serie de caídas sobre un colchón, dentro de las caídas simuladas se encuentra caída de frente, caída de espaldas, caída lateral izquierda y derecha así como caída al recoger un objeto del piso. Para determinar el consumo energético de las aplicaciones desarrolladas se utilizó PowerTutor. Al momento de realizar las pruebas, se mantuvieron condiciones similares en ambos equipos, tales como la desactivación de Bluetooth, GPS, Wi-fi, 3G y NFC. Además solo las aplicaciones relevantes del sistema se dejaron ejecutar así como PowerTutor y la aplicación bajo prueba. Las mediciones energéticas se realizaron en periodos de tiempo iguales para ambas aplicaciones. La Tabla 1 y 2 muestra los resultados alcanzados.

<b>Algoritmo</b>	<b>Consumo de Energía (Joule)</b>	<b>Porcentaje de Uso del CPU</b>
SensorWalk	77.7 J	3.8%
DetectFall2	82.5	4.7%

**Tabla 1. Resultados Obtenidos en el Galaxy S1.**

Para el teléfono Samsung su CPU consumió 77.7J que equivale a 3.8 % del total de la energía utilizada y el teléfono Sony consumió 53.4J que equivale a 2.2 %. También se realizó las mismas pruebas con el segundo algoritmo DetectFall2, donde el CPU del teléfono Samsung consumió 82.5J, que equivale a 4.7% del total de la energía utilizada mientras que el teléfono Sony consumió 46.3J, que equivale al 2.9%. Cada prueba duro 45 minutos en promedio.

<b>Algoritmo</b>	<b>Consumo de Energía (Joule)</b>	<b>Porcentaje de Uso del CPU</b>
SensorWalk	53.4 J	2.2%
DetectFall2	46.3 J	2.9%

**Tabla 2. Resultados Obtenidos en el Sony Xperia S.**

La efectividad de detección de los algoritmos fue de 90% para SensorWalk mientras que DetectFall2 alcanzó un 95% de efectividad. La efectividad de detección se determinó con el número de veces que la alarma se activó cuando se realizó una caída. Las caídas que no pudieron ser detectadas fueron las que no producen un cambio drástico en la señal del acelerómetro, tal como agacharse a recoger un objeto y caerse.

	<b>SensorWalk</b>		<b>DetectFall2</b>	
	Samsung	Sony	Samsung	Sony
<b>Potencia Inicial (mW)</b>	683	687	717	799

<b>Potencia final (mW)</b>	622	544	637	612
<b>Consumo Promedio (mW)</b>	61	143	80	187
<b>Porcentaje</b>	8.9	20.8	11.6	23.4

**Tabla 3. Resultados del Consumo de Potencia.**

Los resultados mostrados en la Tabla 1 y 2, reflejan el promedio general de todas las pruebas realizadas. La tabla de resultados muestra que el porcentaje de uso del CPU y el consumo de energía son más altos en el equipo Samsung que el equipo Sony al probar ambos algoritmos. Esto es debido a la mayor capacidad de procesamiento del CPU del teléfono Sony.

La discrepancia con el valor menor del consumo de energía de la batería, con el algoritmo DetectFall2 contra el algoritmo Sensorwalk, en el teléfono Sony es debido a las estimaciones de consumo de energía son aproximaciones basadas en el modelo de energía de un teléfono diferente como se explicó en el sección 2.3.

Existe un incremento del 2.7% en la cantidad total de potencia utilizada cuando se implementó la aplicación DetectFall2 sobre la implementación de Sensorwalk en el teléfono Samsung. En el equipo Sony hay un incremento 2.6%.

## **4. Conclusiones**

Los teléfonos inteligentes son una excelente plataforma para el desarrollo de algoritmos que permitan que las personas de la tercera edad mantengan su independencia. Estos teléfonos son una plataforma autocontenida que permite tener una serie de sensores y

actuadores junto con la unidad de procesamiento y comunicación. Por lo tanto, se puede realizar la detección de caídas sin necesidad de ningún dispositivo externo. Se demostró que el algoritmo de la detección de caídas por cálculo de derivadas (la aplicación Sensorwalk) es más eficiente en el consumo energético debido a su baja complejidad. Sin embargo, tiene un desempeño por debajo del segundo algoritmo en términos de la efectividad en la detección. Sensorwalk alcanzó un 90% de detección de caídas mientras que DetectFall2 logró detectar exitosamente el 95% de estas. En promedio DetectFall2 consumió solo un 2.5% más de energía, pero alcanzó un 5% más de detecciones positivas. Por lo que DetectFall2 se considera como el algoritmo que más eficiente pues conserva el mejor balance entre efectividad en la detección y consumo de potencia.

## 5. Referencias

- [1] Grupo del Banco Mundial, Población de 65 años de edad y más (% del total), El Banco Mundial, 2009. [En Línea]. [Consultada: 21 Enero 2013] Disponible: <http://datos.bancomundial.org/indicador/SP.POP.65UP.TO.ZS>
- [2] Instituto Nacional de Geriatria, Situación Sociodemográfica del Envejecimiento en México, Instituto Nacional de Geriatria [En Línea]. [Consultada: 21 Enero 2013] Disponible: [http://www.geriatria.salud.gob.mx/descargas/01\\_sit\\_dem\\_env.pdf](http://www.geriatria.salud.gob.mx/descargas/01_sit_dem_env.pdf)
- [3] Organización Mundial de la Salud, Caídas, Centro de prensa de la OMS, Octubre de 2012. [En Línea]. [Consultada: 21 Enero 2013] Disponible: <http://www.who.int/mediacentre/factsheets/fs344/es/index.html>
- [4] X. Yu, "Approaches and Principles of Fall Detection for Elderly and Patient." In Proceedings of 10th IEEE International Conf. On e-Health Networking, Applications and Service (HEALTHCOM 2008), pp. 42-47, 7-9 July 2008

- [5] B. Jansen, y R. Deklerck, "Context Aware inactivity recognition for visual fall detection," *Pervasive Health Conference and Workshops*, pp 1-4, 29 Nov. -1 Dic 2006
- [6] D. Anderson, J. M. Keller, M. Skubic, X. Chen, y Z. He, "Recognizing falls from Silhouettes," 28<sup>th</sup> Int. Conf. of the IEEE Eng. In Medicine and Biology Society, pp 6388-6391, Agosto 2006
- [7] C. Rougier, J. Meunier, A. St-Arnoud, J. Rosseau, "Fall Detection from Human Shape and Motion History using Video Surveillance," 21<sup>st</sup> Int. Conf. on Advanced Information Networking and Application Workshops, Vol. 2, pp. 875-880, 2007
- [8] Lide Zhang, Birjodh Tiwana, Robert P. Dick et al, "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," En las memorias de IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2010, Arizona, USA, pp. 105-114
- [9] PowerTutor, Disponible: <http://powertutor.org> [Consultada: 28 Junio 2013]
- [10] EZ430 Cronos Watch, Disponible:<http://processors.wiki.ti.com/index.php/EZ430-Chronos> [Consultada: 28 Junio 2013]

## **7. Autores**

Ing. Lynette Drusilla Layne es Ingeniero Electrónico por la Universidad Autónoma de San Luis Potosí.

Dra. Ruth M. Aguilar-Ponce es Doctora en Ingeniería en Computación por la Universidad de Louisiana en Lafayette.

Dr. J. Luis Tecpanecatli-Xihuitl es Doctor en Ingeniería en Computación por la Universidad de Louisiana en Lafayette.

M.E.E. Héctor Raúl Moncada González, obtuvo el grado de Maestría en Ingeniería Electrónica con especialidad en bioelectrónica por la Universidad Autónoma de San Luis Potosí (UASLP) en Febrero del 2014.