



**UNIVERSIDAD
DE ANTIOQUIA**

**SISTEMA DE MONITOREO Y ALARMAS PARA
MOVIMIENTO DE ADULTOS MAYORES**

Autor(es)

Sebastian Betancur Ramirez

Universidad de Antioquia

Facultad de ingeniería, Bioingeniería

Medellín, Colombia

2019



SISTEMA DE MONITOREO Y ALARMAS PARA MOVIMIENTO DE ADULTOS MAYORES

Sebastian Betancur Ramirez

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

Bioingeniero

Asesores (a):

Juan Diego Lemos, Ingeniero electrónico

Julian Alvarez Alvarez, Ingeniero de sistemas

Universidad de Antioquia

Facultad de ingeniería, bioingeniería

Medellin, Colombia

2019.

SISTEMA DE MONITOREO Y ALARMAS PARA MOVIMIENTO DE ADULTOS MAYORES

Resumen

El 72% de la población geriátrica en Colombia no puede acceder a un centro especializado de cuidados, por tal motivo se decidió implementar un sistema de monitoreo y alarmas para movimiento de adultos mayores.

Se empezó haciendo un modelo simple del sistema, que constaba de 3 partes, Placa de desarrollo y sensor, almacenamiento en la nube y interfaz con el usuario.

Se decidió trabajar con la Raspberry y el sensor MPU6050 acelerómetro y giroscopio, con protocolo de comunicación i2c, se implementaron unos códigos en Python para captar la señal de 3 ejes del giroscopio y acelerómetro, a las gravedades del acelerómetro se le hizo una transformación para obtener el Angulo de X y Y respecto a Z. Se creo una base de datos en la nube, AWS, para almacenar estas señales captadas y posteriormente ser visualizadas en tiempo real en una página web.

Para la generación de alarmas se creo una tabla donde se almacenaron el tiempo límite en minutos que el paciente puede permanecer en reposo, después de superar este tiempo, se habilita un texto llamativo sugiriendo que se comuniquen con el paciente, adicionalmente se creó una página web para que el médico o algún familiar del paciente pueda cambiar esta alarma autónomamente, también se diseñó una página para que puedan visualizar el histórico de alarmas en cualquier momento. También se diseñó una alarma simulando una caída, esta se activa si en algún momento cualquiera de los 3 ejes presenta una velocidad angular mayor a 100 grados por segundos.

Introducción

Enfrentar la vejez en Colombia se ha convertido en todo un desafío. Según ha concluido un informe de la Facultad de Medicina de la Universidad de La Sabana y la Asociación Colombiana de Gerontología y Geriatria, para 2020 habrá dos adultos mayores por cada adolescente, pero sus condiciones de vida serán preocupantes.

El estudio, que consultó a 30.000 adultos mayores de 250 municipios de Colombia, demostró que el 9,3 % de ellos viven solos, y en Bogotá el 11 % sobreviven en esta condición. Como cualquier ciudadano, los adultos mayores necesitan tener asegurado el ejercicio de su derecho a la salud, sobre todo, porque son propensos a sufrir enfermedades que requieren una atención especializada. Pero el 72 % de

esta población no puede acceder a un centro especializado en cuidados geriátricos ni a un cuidador. (Semana 2017). [1]

Por este motivo, se quiere realizar un sistema de monitoreo para el movimiento de adulto mayores, con el fin de comunicar en todo momento el estado en que se encuentra el paciente, si ha pasado mucho tiempo sin moverse o ha sufrido alguna caída. Esto se logrará integrando un sistema embebido con servicios de internet de las cosas.

La atención médica y el cuidado de la salud representan una de las áreas de aplicación más atractivas para la IoT. Internet de las Cosas tiene el potencial de dar lugar a muchas aplicaciones médicas tales como la monitorización remota de la salud, programas de acondicionamiento físico, enfermedades crónicas o atención de la tercera edad. El cumplimiento del tratamiento y la medicación en casa vigilado por profesionales de la salud es otra aplicación potencial importante. Por lo tanto, los sensores y dispositivos de diagnóstico y de imágenes médicas pueden ser vistos como dispositivos inteligentes u objetos que constituyen una parte fundamental de IoT. Se espera que los servicios de salud basados en Internet de las Cosas puedan reducir costes, aumentar la calidad de vida, y mejorar la experiencia del usuario. [2]

Accenture tiene un conjunto de ofertas llamado Connected Health Solutions, aprovechan las tecnologías de IoT para optimizar los procesos existentes y habilitar a los pacientes, proveedores y pagadores dentro de la industria del cuidado de la salud y las ciencias de la vida. El ecosistema de IoT incluye: terminales digitales como dispositivos y sensores, diversas plataformas en la nube, análisis y bases de datos. Las áreas de enfoque incluyen: servicios para pacientes, análisis predictivo, ensayos clínicos, cuidado preventivo, fabricación inteligente, gestión de dispositivos, bienestar y prevención, sin embargo, este servicio aún no se ofrece en Colombia. [3, 4, 5]

Accenture está muy interesado en realizar este proyecto porque cumple todas las características de Connected Health Solutions, además, Medellín se convertirá en la sede principal de Accenture Technology en Latinoamérica y tiene que ser pioneros en proyectos de este tipo para poder empezar a brindar todos sus servicios.

Objetivos

Diseñar, implementar y probar sistema para monitoreo de adultos mayores usando servicios de internet de las cosas.

Objetivo Específicos

- Analizar y seleccionar un proveedor de servicios de internet de las cosas (IoT), teniendo en cuenta las necesidades del proyecto, y las alianzas y asociaciones de Accenture con estos proveedores.
- Analizar y elegir placa de desarrollo y lenguaje de programación que se adapte a las necesidades del proyecto.
- Seleccionar y analizar el comportamiento de diferentes tipos de sensores, giroscopio y acelerómetro que pueden ser usados en el desarrollo de la aplicación.
- Diseñar e implementar un prototipo del sistema, y diseñar y realizar un conjunto básico de pruebas funcionales del mismo.

Marco Teórico

Internet de las cosas (IoT)

El internet de las cosas, IoT, se refiere a la interconexión digital de los objetos cotidianos con internet, conformado de múltiples tecnologías como sensores que permiten conectar el mundo físico con el digital, computadores que permiten procesar esa información y plataformas web donde se procesan y almacenan los datos. [8]

Los principales proveedores de servicios de internet de las cosas son:

- Amazon Web Services.
- Microsoft Azure.
- Sap Leonardo
- Siemens Mindsphere

Sistema embebido

Un Sistema Embebido es un sistema electrónico diseñado para realizar pocas funciones en tiempo real, según sea el caso. Al contrario de lo que ocurre con las computadoras, las cuales tienen un propósito general, ya que están diseñadas para cubrir un amplio rango de necesidades y los Sistemas Embebidos se diseñan para cubrir necesidades específicas. [7]

Giroscopio

Un Giroscopio es un pequeño dispositivo que sirve para medir la gravedad de la tierra, esto a su vez ayuda a determinar la orientación del móvil. Su diseño consiste en un disco de giro libre el cual se denomina Rotor, este va montado sobre un eje de giro en el centro de una rueda más estable y grande. A medida que el eje gira, el Rotor permanece estacionario indicando así la fuerza de gravedad en el centro.

Acelerómetro

Un acelerómetro, como su nombre lo indica, es un dispositivo electromecánico que nos sirve para medir las fuerzas de aceleración, ya sea estática o dinámica. Son útiles para detectar las vibraciones y el movimiento en los sistemas. Son comúnmente usados para activar sistemas de protección en vehículos, o computadoras, para registrar la actividad física en las personas, para aplicaciones de orientación en los dispositivos móviles, etc.

Lenguajes de programación:

Python

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. [8]

C++

La definición "oficial" del lenguaje nos dice que C++ es un lenguaje de propósito general basado en el C, al que se han añadido nuevos tipos de datos, clases, plantillas, mecanismo de excepciones, sistema de espacios de nombres, funciones inline, sobrecarga de operadores, referencias, operadores para manejo de

memoria persistente, y algunas utilidades adicionales de librería (en realidad la librería Estándar C es un subconjunto de la librería C++). [9]

HTML

HTML, que significa Lenguaje de Marcado para Hipertextos (HyperText Markup Language) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad. Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript).[10]

PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

Metodología

Tomando como referencia la definición de internet de las cosas, lo primero que se hizo fue diseñar un modelo sencillo, figura 1, para el sistema de monitoreo y alarmas para movimiento de adultos mayores.

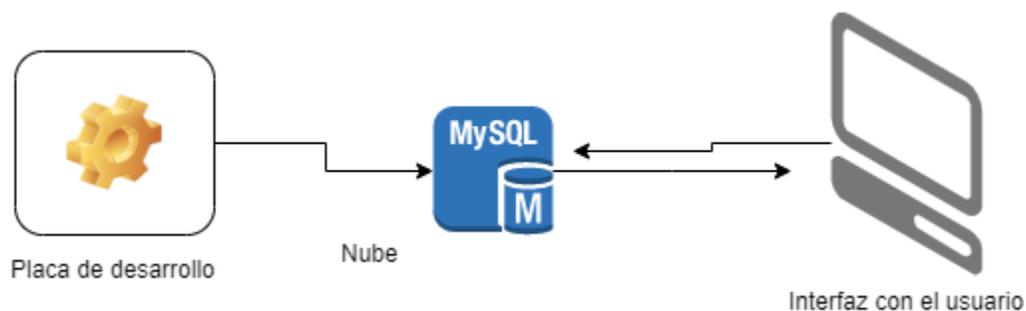


Figura 1. Esquema general del sistema.

Para ello se necesitó: un sensor, giroscopio y acelerómetro, conectado a una tabla de desarrollo, guardando la información en una base de datos en la nube para finalmente poder ser vista por el usuario en una página web.

Después de tener listo el diseño del modelo, surgieron 3 preguntas que necesitaban ser resueltas para poder darle solución al proyecto.

¿Qué proveedor de servicios de internet de las cosas voy a emplear?

¿Qué placa de desarrollo y lenguaje de programación voy a usar?

¿Qué sensor voy a seleccionar?

Para darle solución a la primera pregunta, se investigaron las alianzas que tenía Accenture y que proveedores empleaba en sus proyectos o en los de sus clientes, también se hizo una comparación analizando ventajas y desventajas de cada uno. Después de tener seleccionado el proveedor de servicios de internet de las cosas, se realizó una capacitación de creación y mantenimiento de bases de datos en la nube.

Continuando con la siguiente pregunta, para seleccionar la placa de desarrollo, se hizo una comparación entre los dos microcontroladores con los que había trabajado en proyectos universitarios, Raspberry y PsocCreator, se tuvo en cuenta el tamaño de estas, precio, dinamismo con lenguajes de programación, conectividad, bibliografía y referencias.

Para seleccionar el lenguaje de programación se hizo una comparación teniendo en cuenta la simplicidad de código, compilación, bibliotecas y rutinas y la moda del mercado.

Así mismo sin saber que sensor usar, se hizo una tabla comparando los girómetros y osciloscopios más usados, tomando en cuenta sus ejes, rango, protocolos de comunicación, requerimientos de energía y características.

Luego de tener las 3 preguntas resueltas, se continuo con la implementación del sistema de monitoreo para adultos mayores realizando la siguiente secuencia de pasos:

- Se instaló el sistema operativo Raspbian a la Raspberry: Lo primero que se hizo fue formatear una tarjeta microSD de 32 Gb, luego se descargó en la MicroSD un archivo llamado NOOB de la página oficial de Raspberry. Se colocó la MicroSD en la tarjeta, se conectó un cable Ethernet y se instaló el sistema operativo.
- Luego de tener el sistema operativo listo, se realizó la conexión de la Raspberry a wifi, esto con el fin de comodidad y portabilidad.
- Se buscó bibliografía del funcionamiento del sensor MPU6050 y se hizo la conexión a la Raspberry, como se muestra en la figura 2.

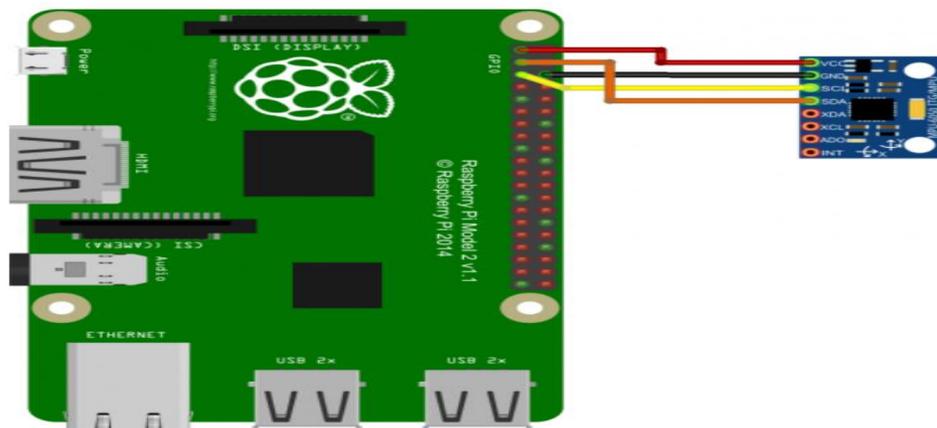


Figura 2. Conexión RaspBerry y sensor MPU6050.

- Se configuró la Raspberry para la lectura I2C, para ello se habilitó la entrada por I2C e ingresó en la interfaz de comandos la siguiente línea de código:
`sudo apt-get install i2c-tools python-smbus.`
- Se diseñó y escribió código en Python para acondicionamiento de la señal del giroscopio y del acelerómetro, para ello se usaron las librerías `smbus` y `math`.
- Después de tener lista el acondicionamiento de la señal, se realizó una capacitación para crear base de datos MySQL en la nube de Amazon web services.
- Se creó una tabla para el almacenamiento de los registros generados por el sensor.
- Se escribió código en Python para conectar Raspberry a la base de datos, para ello se importó la librería MySQL.

- Se diseño Script para añadir registros con la información generada por el sensor.
- Se diseño página web para visualización del monitoreo, adicionalmente se escribió código para conexión de la pagina web con la base de datos.
- Se implemento Query para extraer registro de la base de datos.
- Se diseño y creo página para la generación de alertas.
- Se creo tabla para guardar Histórico de las alertas.
- Se diseño página web para visualizar el histórico de las alertas.

Resultados y análisis

Analizando los proyectos de Accenture y de sus clientes, se evidencio que se empleaba a dos proveedores líderes, Azure y AWS. Con esta información se realizo una tabla de ventajas y desventajas, tabla 1. Se comparo y selecciono el proveedor que mejor se ajustó al proyecto.

Tabla 1. Comparación entre AWS y Azure.

Proveedor	Ventajas	Desventajas
AWS	<ul style="list-style-type: none"> -Socio de Accenture. -AWS Training and Certification. -Bajo costo. -Factura por lo que se utiliza. -Es escalable, -Permite trabajar con todo tipo de bases de datos - Seguro, porcentaje de error en disponibilidad y/o pérdida de datos del 0,000001%. 	-No apto para amateurs.
AZURE	-Bajo costo, 5 veces menos en	-Aplicaciones más lentas.

	almacenamiento SQL que AWS. -Seguro. -Ofertas y promociones. - 30 años de experiencia. -90 certificaciones.	
--	--	--

Al ver los resultados de la tabla 1, se decidió trabajar con Amazon web Services, las razones principales por las que se prefirió este proveedor fueron: su asociación con Accenture, tenemos acceso al menos 3 capacitaciones en línea mensuales sin ningún costo en AWS Training and Certification y la posibilidad de trabajar con todo tipo de bases de datos.

Para seleccionar la placa de desarrollo se tuvieron en cuenta 2 placas, basado en la experiencia que tenía, se decidió hacer una comparación entre RaspBerry y PScoc, ver tabla 2.

Tabla 2. Comparación placas de desarrollo. [11]

Placa	Precio en pesos.	Tamaño.	Lenguajes de programación	Ventajas.	Desventajas.
RaspBerry	180.000	Pequeño	-Python -C++	-Mucha bibliografía. -Conexión a internet. -Simplicidad en códigos. -Comunicación I2C, SPI, UART.	-Necesita sistema operativo. -Se recalienta.

Psoc	120.00 0	Mediano	-C++	-Bastante experiencia. -Muchos pines E/S. -Bajo consumo de energía. - Comunicación I2C, SPI, ART, LIN, I2S.	-Escasa bibliografía. -Mala comunicación con sistemas exteriores.
------	-------------	---------	------	--	--

Teniendo en cuenta la comparación realizada en la tabla 2, se decidió trabajar con Raspberry porque sus características se ajustaban mas al proyecto, tiene un tamaño pequeño, permite realizar conexión con el exterior, en este caso bases de datos, permite comunicación con sensores digitales i2c y SPI, es programable en varios lenguajes y existe mucha bibliografía lo que hace mas sencillo el desarrollo del proyecto.

Se decide trabajar con Python porque es un lenguaje sencillo, compilación simple, tiene librerías para trabajar con i2c, además también tiene librerías para trabajar con bases de datos, administración de pines E/S, así mismo funciona en muchos sistemas operativos y hay mucha bibliografía y referencia.

Se realizo la comparación de 10 sensores distintos, como se observa en la tabla 3. Se hizo la comparación de sensores analógicos y digitales, de solo girómetros y de girómetros y acelerómetros integrados.

Tabla 3. Comparación de los Giroscopios y acelerómetros más usados. [12]

Dispositivo	Ejes	Rango	Interfaz	Requerimientos de energías	Características
IDG1215	2 (X, Y)	$\pm 67^\circ/s$	Analógica	3V100 μ A	-Incluye amplificador y filtro paso bajo.

LPR503AL	2 (X, Y)	$\pm 30^\circ/s \pm 120^\circ/s$	Analógica	2.7V a 3.6 V 6.8 mA	- Filtro paso bajo. - SEFL-TEST. - Power down.
LPR5150AL	2 (X, Y)	$\pm 1500^\circ/s \pm 6000^\circ/s$	Analógica	2.7V a 3.6 V 6.8 mA	- Filtro paso bajo. - SEFL-TEST. - Power down.
ITG3200	3 (X, Y, Z)	$\pm 2000^\circ/s$	Digital I2C-16 bits	2.1V a 3.6V 6.5 mA	- Filtro paso bajo programable. - Sensor de temperatura.
MPU-6050	3 acelerómetros (X, Y, Z) 3 girómetros (X, Y, Z)	$\pm 250, \pm 500, \pm 1000,$ and $\pm 2000^\circ/sec \pm$ $2g, \pm 4g, \pm 8g$ and $\pm 16g$	Digital I2C	2.375V- 3.46V 3.6 mA- 500 μ A	- Filtro paso bajo programable. - Detección de orientación. - Interrupciones programables. - Detección de caída libre. - SEL-TEST. - Detección de golpe.
LPY503AL	2 (X, Z)	$\pm 30^\circ/s \pm 120^\circ/s$	Analógica	2.7 V a 3.6 V 6.8mA	- Filtro paso bajo. - SEFL-TEST. - Power down.
MPU-3050	3 (X, Y, Z)	$\pm 250, \pm 500, \pm 1000,$ and $\pm 2000^\circ/sec$	Digital I2C	2.1V a 3.6V 6.1 mA	- Interrupciones programable soporta características tales como el reconocimiento de gestos, paneo,

					detección de cero movimientos, detección de golpe y sacudida. - Filtro paso bajo programable. - Funcionalidad podómetro.
MPU-6000	3 acelerómetros (X, Y, Z) 3 girómetros (X, Y, Z)	$\pm 250, \pm 500, \pm 1000,$ $\text{and } \pm 2000^\circ/\text{sec} \pm 2g,$ $\pm 4g, \pm 8g \text{ and}$ $\pm 16g$	Digital SPI	2.375V- 3.46VG-3.6 mA- 500 μ A	- Filtro paso bajo programable. - Detección de orientación. - Interrupciones programables . - Detección de caída libre. - SEL-TEST. - Detección de golpe
MAX-21000	3 (X, Y, Z)	$\pm 31.25/\pm 62.50/\pm 125/$ $\pm 250/\pm 500/\pm 1k/\pm 2k$ $^\circ/s$	Digital I2C, SPI	1.71V a 3.6V 5.4 mA	-Filtro paso bajo programable. (Interrupciones programables)
BMG160	3 (X, Y, Z)	$\pm 125^\circ/s \text{ a } \pm 2000^\circ/s$	Digital I2C, SPI	2.4V a 3.6V 5mA	- Filtro paso bajo. - Sensor de temperatura.

Observando la tabla 3, se decidió trabajar con el sensor MPU6050 porque es un sensor que está compuesto por 3 acelerómetros y 3 giróscopos, cada uno con su respectivo ADC de 16 bits. Presenta la posibilidad de modificar la escala de trabajo,

tabla 4, proporcionando así más precisión para movimientos más lentos. Además, tiene los requerimientos de energía más bajos, asimismo trabaja con protocolo I2C, ideal para trabajar con Raspberry y Python, también tiene mejores características que el resto de los sensores.

Se decidió trabajar con la escala de ± 2 para más precisión.

Tabla 4. Rango de escalas.

Acelerómetro	$\pm 2g$	$\pm 4g$	± 8	$16\pm$
Giroscopio	$\pm 250^\circ/\text{seg}$	$\pm 500^\circ/\text{seg}$	$\pm 1000^\circ/\text{seg}$	$\pm 2000^\circ/\text{seg}$

Con todo definido para el sistema, se implementó un código en Python para obtener los ángulos del eje X y del eje Y, y la velocidad angular de los 3 ejes, X, Y y Z.

Adicionalmente se implementó un código de conexión con la base de datos, cada 300 milisegundos se le agregaba un registro. Para ello se utilizaron las librerías time, mysql, smbus y math.

Para almacenar esta información se creó una tabla, figura 3, con los siguientes campos:

Índice, Angulo en X, Angulo en Y, Velocidad angular en X, Velocidad angular en Y, Velocidad angular en Z.

Índice	GradosX	GradosY	GiroX	GiroY	GiroZ
1	3.000000	0.000000	4.000000	45.000000	56.000000
2	2.300000	1.200000	0.080000	0.900000	0.080000
3	0.300000	0.200000	0.080000	0.900000	0.080000
4	0.300000	0.200000	0.080000	0.900000	0.080000
5	0.300000	0.200000	0.080000	0.900000	0.080000

Figura 3. Tabla Sensor, variables del sensor.

Después de tener la información de los ejes y sus velocidades en la web, se procedió a diseñar una página web para poder visualizarlas, como se muestra en la figura 4, se observa el título del proyecto, 1 botón para cambiar alarma, otro

botón para ver el registro de todas las alarmas que se han generado, y la información que nos brinda el sensor.



Figura 4. Pagina inicial, visualización de las variables de monitoreo.

Para obtener la información de la tabla sensor, se realizó un código php de conexión con la base de datos, también se realizó un código php y html, que ejecuta un query que retorna el ultimo registro guardado en la tabla, este query es generado cada 300 ms, a la misma velocidad que el código en Python me añade un registro.

Como observamos en la figura 4, hay un botón que se llama "Cambiar alarma", cuando damos click en este botón nos abre una nueva ventana, ver figura 5, esta página lo que hace es cambiar la alarma de tiempo en minutos, esto con el fin de llevar el control de cuanto es el tiempo máximo que el individuo puede permanecer sin moverse.

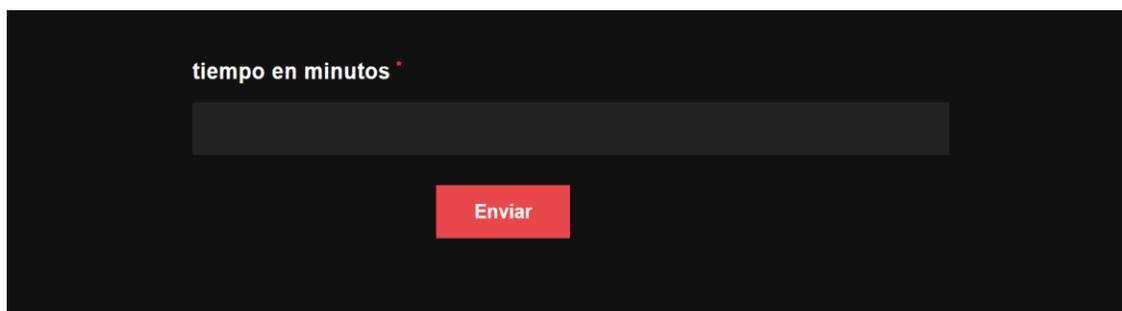


Figura 5. Modificación de tiempo de alarma.

Cuando damos click en Enviar, se añade un registro a una tabla, figura 6, que guarda el histórico de alarmas, esto es gracias a un script Sql, Insert, diseñado dentro de un código php.

También en la figura 4, podemos ver un botón que se llama "ver alarmas", cuando damos click a este botón nos abre una nueva ventana, figura 5, donde podemos visualizar una tabla del registro histórico de alarmas.

----Conectado a la base de datos

Indice -	Tiempo de alarma en minutos -	Fecha -	Hora -
1	5	2019-10-15	03:34:53
2	2	2019-10-15	03:39:03
3	4	2019-10-15	03:39:25
4	1	2019-10-15	04:04:52
5	1	2019-10-15	04:15:33

Figura 6. Histórico de alarmas.

Se guarda este histórico con el fin de llevar un control mas detallado del cambio de la alarma de tiempo, un médico o un familiar tienen la autonomía de cambiar esta alarma para poder estar mejor comunicados con el estado del paciente.

Hay dos maneras posibles de que se active la alarma, que el paciente este cierto tiempo en reposo, sin moverse, o que la velocidad angular de cualquiera de los ejes haya superado un valor específico, en este caso 100°/seg.

Para la alarma de reposo, se verifica que el eje X ni el Eje Y tenga cambio en su ángulo más de 20 grados en el tiempo especificado, para esta prueba se estableció un tiempo de 1 minuto para ver el correcto funcionamiento de este.

Cuando la alarma se activa, ver figura 7, en la página web aparece un botón en rojo informando que se deben comunicar con el paciente.



Figura 7. Alarma activada.

Como se puede ver en la figura 7, el giro en X y en Z superaban los 100/seg, esto fue lo que me activo la alarma, esto se realizó con el fin de simular la caída del paciente.

Conclusiones:

- Se decidió trabajar con Amazon web Services, las razones principales por los que se prefirió este proveedor fueron: su asociación con Accenture, acceso al menos 3 capacitaciones en línea mensuales sin ningún costo en AWS Training and Certification y la posibilidad de trabajar con todo tipo de bases de datos.

-Se decidió trabajar con Raspberry porque sus características se ajustaban más al proyecto, tiene un tamaño pequeño, permite realizar conexión con el exterior, conexión con bases de datos, permite comunicación con sensores digitales i2c y SPI, es programable en varios lenguajes y existe mucha bibliografía en donde se pudo apoyar.

-Se decidió trabajar con Python porque es un lenguaje sencillo, compilación simple, tiene librerías para trabajar con i2c, tiene librerías para trabajar con bases de datos, administración de pines E/S, y el SO operativo de Raspberry lo trae por defecto.

-Se diseño un prototipo del sistema, el conjunto de pruebas que se realizo fue muy básico, el objetivo principal de estas fue probar la funcionalidad de este.

-Para poder desarrollar el proyecto se necesitaron amplios conocimientos en sensores digitales y competencias altas en programación, se hicieron en desarrollos en Python, Sql, php, html y css.

-Se pensó en algún momento en hacer el prototipo portátil, sin embargo no se pudo porque se encontraron varias limitantes como el consumo de energía de la raspberry, la conexión con el sensor, este tiene huecos muy grandes que no permitieron buena conectividad y se necesitó una protoboard.

Referencias Bibliográficas

[1]. (2017) Ser adulto mayor en Colombia, todo un karma, *Revista Semana*, tomado el 19/08/2018 del enlace web: <https://www.semana.com/vida-moderna/articulo/calidad-de-vida-de-los-adultos-mayores-en-colombia-2017/527154>

[2]. (2017) Mejora de salud y vida con IoT, *the new now*, tomado el 19/08/2018 del enlace web: <https://www.thenewnow.es/innovacion/mejora-salud-calidad-vida-iot/>

[3]. CONNECTED HEALTH AROUND THE GLOBE, *Accenture*, tomado el 19/08/2018 del enlace web: <https://www.accenture.com/us-en/insight-making-case-connected-health>

[4]. CONNECTED HEALTH, *Accenture*, tomado el 19/08/2018 del enlace web: <https://www.accenture.com/us-en/service-connected-health>

[5] Connected Health Solutions, *Accenture*, tomado el 19/08/2018 del enlace web: <https://in.accenture.com/internetofthings/connected-health-solutions/>

[6]. Ingeniería asistida por computador, tomado el 20/08/2018 del enlace web: <http://www.iac.com.co/que-es-iot/>

[7]. Infotec, SemanticWebBuilder, tomado el 20/08/2018 del enlace web: http://www.semanticwebbuilder.org.mx/es_mx/swb/Sistemas_Embebidos_Innovando_hacia_los_Sistemas_Inteligentes_

[8]. El tutorial de Python, Guido van Rossum, tomado el 20/07/2019 del enlace web: <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>

[9]. Manual de Programación en Lenguaje C++, Linda I. Olivares Flores, UNAM, 2008. Tomado el 20/07/2019 del enlace web:

<https://paginas.matem.unam.mx/pderbf/images/documentos/mprogintc%2B%2B.pdf>

[10]. Curso completo de HTML, Jorge Ferrer, Víctor García, Rodrigo García, tomado el 20/07/2019 del enlace web:

<http://es.tldp.org/Manuales-LuCAS/doc-curso-html/doc-curso-html.pdf>

[11] INTRODUCCIÓN AL PSoC5LP Teoría y aplicaciones prácticas, Julián Rolando Camargo López, César Andrey Perdomo Charry, Universidad Distrital Francisco José de Caldas, tomado el 01/10/2019 del enlace web:

<https://editorial.udistrital.edu.co/contenido/c-877.pdf>

[12] Introduccion al giroscopio, 5 HERTZ Electronica, tomado el 10/09/2019 del enlace web:

https://www.5hertz.com/index.php?route=tutoriales/tutorial&tutorial_id=13

Anexos

1. Acelerometro.py
2. index.php
3. conexión.php
4. insertar.php
5. mostrar_tabla.php
6. pedidos.html
7. Principal.css
8. Boton.css