



DESARROLLO DE UN PROTOTIPO DE SISTEMA DE CAPTURA DE  
MOVIMIENTO PARA ACTIVIDAD FÍSICA DEL MIEMBRO INFERIOR  
COMO INTERFAZ DE USUARIO EN UN AMBIENTE DE REALIDAD  
VIRTUAL

Universidad Militar Nueva Granada Facultad de Ingeniería  
Ingeniería en Mecatrónica  
Bogotá D.C., Colombia  
2015

DESARROLLO DE UN PROTOTIPO DE SISTEMA DE CAPTURA DE  
MOVIMIENTO PARA ACTIVIDAD FÍSICA DEL MIEMBRO INFERIOR  
COMO INTERFAZ DE USUARIO EN UN AMBIENTE DE REALIDAD  
VIRTUAL

Estudiante:  
Jennifer Estefanía Ramos  
Montilla

Modalidad: Auxiliar de  
Investigación

Código:  
1801394

Líder de Proyecto de  
Investigación ING 1545:  
Alvaro Uribe Eng.D.

Universidad Militar Nueva Granada Facultad de Ingeniería  
Ingeniería en Mecatrónica  
Bogotá D.C., Colombia  
2015

## **DEDICATORIA**

De corazón dedico mi trabajo de investigación a quienes de corazón me entregaron...

Para ti Padre Todopoderoso que me diste la vida y con ella el amor, la salud y la fuerza necesaria para no decaer cuando las adversidades se presentaran, aprendiendo a sacar lo mejor de cada situación y a orientar mis objetivos a contribuirle en la mejor forma posible a nuestra sociedad

A mi madre y a mi padre por su amor incondicional, su apoyo, consejos, comprensión, valores, principios inculcados y por sus esfuerzos para lograr mis objetivos,

A mis hermanos por estar siempre presente en mi vida sirviendo de apoyo en todo momento, a mi familia por confiar en mí, impulsarme siempre y recordarme que los sueños que me propusiera los podía alcanzar.

A todos aquellos que de una u otra forma me acompañaron en el transcurso de mi carrera, haciéndome vivir experiencias agradables e inolvidables que aportaron conocimientos y vivencias en mi crecimiento personal y espiritual.

## **AGRADECIMIENTOS**

Gracias Dios por darme esta vida llena de amor, salud, alegrías y también por las dificultades y tristezas porque a través de ellas me he fortalecido y ha crecido mi ser.

Gracias padres por tanto amor convertido en comprensión, paciencia, esfuerzos para apoyarme en todo momento de mi vida,

Gracias hermanitos y amor mío, su presencia ha sido parte fundamental para progresar, tomar la vida con serenidad y hacerla más amable.

Gracias a los docentes, formadores y en general a la Universidad que me transmitieron sus conocimientos y me infundieron valores y principios para contribuir de una mejor forma a nuestra sociedad, en especial a mi tutor Alvaro Joffre Uribe Quevedo quien me guio, apoyó y asistió con perseverancia durante el desarrollo de la presente investigación.

Gracias a mi familia y a todos los que han creído en mí y me impulsaron a seguir adelante haciendo presencia siempre tanto en las situaciones buenas como en las difíciles.

**GRACIAS.....**

## RESUMEN

La actividad física asistida por fisioterapeutas, terapeutas ocupacionales o incluso entrenadores físicos presenta oportunidades de mejora a través de desarrollos tecnológicos e investigación aplicada que pueden resultar en la optimización de estos procesos. Ejercitarse a diario es de vital importancia y trae numerosos beneficios, el más conocido, mantenernos en el peso ideal, al realizar actividad física reducimos el riesgo de padecer enfermedades, mantenemos flexibilidad en articulaciones tendones y ligamentos. La captura de movimiento permite cuantificar el progreso del usuario y proveer un medio de supervisión complementario a medios tradicionales basadas en encuestas y observación. Una ventaja adicional que se presenta al incluir tecnologías en estos procesos es que la masificación y bajos costos de varios dispositivos de interfaz de usuario 3D, permiten crear soluciones flexibles con potencial uso desde casa, en viajes o distintos lugares sin incurrir en gastos de desplazamiento o equipos avanzados permitiendo ampliar la cobertura y acceso, ya que la información adquirida puede ser enviada a especialistas para su estudio y análisis.

Para el desarrollo de este proyecto, se utilizó un sensor inercial que permite la captura de datos, como, la rotación de una articulación a partir de la información del acelerómetro, giroscopio. Esta información se articula como medio de interacción con un ambiente tipo juego, que ofrece entretenimiento al usuario mientras realiza el movimiento del miembro inferior. Para validar la implementación de este dispositivo de captura de movimiento se realizaron pruebas para verificar los datos capturados y la motivación generada en los usuarios.

**Palabras clave: Captura de Movimiento, Ejercitación, Juego,**

## **ABSTRACT**

Assisted physical activity presents improvement opportunities regarding assessment and follow-up. Daily exercising is vital to keep a good health by maintaining proper weight and reducing the possibilities of suffering musculoskeletal disorders. Motion capture allows quantifying exercises and allows monitoring and better assessments as a complementary tool to traditional means. Current advances in 3D user interfaces have resulted in massive use of such devices as motion capture tools that are resulting in flexible tools to improve exercising.

For the development of this project an inertial sensor was used, from the collected data it was possible to calculate the lower limb rotation. The captured information was used as suitable inputs to an exergame virtual environment to promote and capture lower limb exercising. The motion capture device was validated through a test to assess the validity of motion capture data and interest on users to use the solution

**Keywords: Exercising, Game, Motion Capture**

## TABLA DE CONTENIDO

CAPÍTULO 1: INTRODUCCIÓN.....	10
1.1.    Presentación del problema .....	10
1.2.    Justificación .....	10
1.3.    Objetivos .....	11
1.3.1.    Objetivo General .....	11
1.3.2.    Objetivos Específicos.....	11
1.4.    Presentación del documento .....	12
CAPÍTULO 2: ESTADO DEL ARTE .....	13
2.1.    Historia del Exergame .....	14
2.2.    Antecedentes de los juegos serios para la salud.....	15
2.3.    Tecnologías enfocadas a la salud .....	16
CAPÍTULO 3: DESARROLLO .....	19
3.1.    Caracterización del miembro inferior y sus movimientos asociados.....	19
3.1.1.    Miembro inferior.....	20
3.1.2.    Articulaciones de la pierna.....	20
3.2.    Biomecánica del miembro inferior - Cinemática.....	21
3.2.1    Cinemática Directa .....	22
3.3.    Captura de Movimiento .....	23
3.4.    Arquitectura del sistema.....	24
3.5.    Implementación de la captura de movimiento y transmisión de datos .....	25
3.5.1.    Implementación de captura de movimiento.....	25
3.5.2.    Implementación de transmisión de datos.....	29
3.6.    Diseño del Aplicativo .....	35
CAPÍTULO 4: RESULTADOS .....	40
4.1 Dispositivo obtenido .....	40
4.2 Interfaz de usuario .....	41
4.3 Validación .....	42
4.3.1    Encuestas.....	47
CAPÍTULO 5: CONCLUSIONES .....	50

Bibliografía .....	52
Apéndice 1 Conexión Arduino – Sensor Inercial .....	56
Apéndice 2 Configuración comandos AT .....	58
Apéndice 3 Conexión Arduino - sensor Inercial - Bluetooth .....	60
Apéndice 4 Processing: Recepción de datos .....	64
Apéndice 5 Processing: Interfaz y juego .....	66



## LISTA DE FIGURAS

Figura 1 Planos de movimiento del cuerpo humano (tomado de: (Referencia)).....	19
Figura 2 Sistema de soporte y actuación .....	20
Figura 3 Amplitud de la flexión/extensión de la pierna .....	21
Figura 4 Modelo propuesto para simulación cinemática.....	22
Figura 5 Ubicación del sensor.....	23
Figura 6 Arquitectura del sistema propuesto .....	25
Figura 7 Grados de Libertad del sensor asociado a los planos de movimiento .....	26
Figura 8 Circuito de conexión MPU9150.....	26
Figura 9 Diagrama de flujo, programa de conexión con MPU9150. ....	28
Figura 10 Algoritmo de Madgwick .....	29
Figura 11 Datos obtenidos del sensor, movimientos: Flexión – Extensión, Aducción – Abducción. ....	29
Figura 12 Circuito de conexión con el modulo Bluetooth.....	30
Figura 13 Diagrama de flujo, Programa de configuración con comandos AT. ....	32
Figura 14 Monitor Serial, Configuración módulo Bluetooth.....	32
Figura 15 Diagrama de flujo, Programa conexión a sensor MPU9150 y sensor Bluetooth. ....	33
Figura 16 Diagrama de flujo, Programa de recepción de datos en Processing.....	34
Figura 17 Processing 2.2.1, Datos de posición transmitidos.....	35
Figura 18 Primeros ambientes .....	37
Figura 19 Diagrama de flujo, Interfaz-Juego. ....	39
Figura 20 Dispositivo obtenido. ....	40
Figura 21 Ubicación del dispositivo.....	40
Figura 22 Nuevo accesorio .....	41
Figura 23 Interfaz de usuario en Processing .....	41
Figura 24 Opción reporte. ....	42
Figura 25 Validación de datos del acelerómetro contra el instrumento de medición tipo goniómetro en movimiento .....	47

## LISTA DE TABLAS

Tabla 1 Grupos musculares .....	20
Tabla 2 Parámetros cinemática directa.....	22
Tabla 3 Comandos AT modulo Bluetooth HC-05.....	30
Tabla 4 Comparación de datos.....	46

# CAPÍTULO 1: INTRODUCCIÓN

El acrónimo exergames, cuya tracción en español aún no existe y hace referencia a un nuevo concepto de juegos en el cual se promueve la actividad física en el estilo de vida de las personas. Para que los video-juegos se encuentren dentro del conjunto que encierra este concepto, deben basarse en tecnologías que permitan hacer una retroalimentación o seguimiento de movimiento o reacciones del jugador [1].

Una de las aplicaciones de los exergames se enfoca en el sector del cuidado de la salud, entrando a un nuevo campo, en el que juegos serios se enfocan en rehabilitación física de los pacientes, donde la interacción de usuario con el videojuego es esencial para monitoreo y motivación [2].

## 1.1. Presentación del problema

El desarrollo de actividades físicas para el miembro inferior se concentra en la realización de movimientos orientados a fortalecer, acondicionar o recuperar la movilidad en caso de alguna afección o trauma. El desarrollo y práctica de actividad física se caracteriza por estar compuesta de movimientos repetitivos cuya ejecución y éxito depende la motivación intrínseca del usuario, ya que requieren de un alto grado de compromiso para cumplir la meta y cuyo resultado es un beneficio. Este proceso se ve afectado cuando el usuario pierde la motivación por distintos factores como unas pobres guías, malas indicaciones, cansancio, dolor, pereza, falta de tiempo, o simplemente no le da la importancia necesaria [3]. Debido a que en muchas ocasiones las secuencias de ejercicios pueden ser realizadas sin ayuda y sin supervisión, sin ayuda y con supervisión, con ayuda y con supervisión, o con ayuda y sin supervisión, es en estos escenarios donde el usuario ni siquiera cuenta con la realimentación del especialista y queda realizando los ejercicios por cuenta propia según lo que considere es una buena ejecución. En la actualidad esta falta de cuantificación puede ser subsanada debido sensores y dispositivos de captura de movimiento, que permitirían seguir más cerca el comportamiento del paciente.

## 1.2. Justificación

Según la OMS<sup>1</sup> (Organización Mundial de la Salud), sugiere que, más de mil millones de personas de la población mundial experimentan algún tipo de discapacidad, esta cifra aumenta conforme la

---

<sup>1</sup>[http://www.who.int/disabilities/world\\_report/2011/en](http://www.who.int/disabilities/world_report/2011/en)

población crece, el envejecimiento y enfermedades crónicas se manifiestan. Dentro de las mayores causas de discapacidad se tienen las enfermedades crónicas como diabetes, enfermedad cardiovascular y cáncer; heridas como aquellas resultantes de un accidente, conflictos, caídas, minas, desórdenes mentales, defectos de nacimiento, malnutrición, HIV y SIDA entre otras. En Colombia según estudios realizados por el DANE<sup>2</sup> en el 2010, de 570. De 306 personas encuestadas, la mayor causa por la cual las personas no reciben servicios de actividad física relacionada con rehabilitación es por la falta de dinero seguida por desinformación, en algunos casos al termino de las sesiones, el usuario cree que ya no la necesita, otra causa es la lejanía del centro de rehabilitación, en algunos casos no les gusta o no hay quien los lleve.

Este tipo de alteraciones en el movimiento afectan la realización de labores cotidianas como alimentarse, asearse, vestirse, cambiar de postura y mantenerla, desplazar y relacionarse con objetos y personas, caminar, correr o saltar. Adicional a las razones para no rehabilitarse y las consecuencias generadas por la alteración de la movilidad, los procesos fisioterapéuticos se caracterizan por la ejecución de movimientos repetitivos que cuando son realizados bajo supervisión, una desventaja es que la validación sobre la evolución puede ser subjetiva y relativa a la experiencia y destreza del fisioterapeuta [4].

La Organización Mundial de la Salud<sup>3</sup> reconoce la importancia de la tecnología para solucionar problemas de salud y mejorar la calidad de vida de las personas. La OMS sostiene que las tecnologías son un componente que permite prevenir, diagnosticar y tratar enfermedades aliviando deficiencias o discapacidades. En 2005 la OMS adoptó la resolución e-Health como esfuerzo para fortalecer los sistemas de salud en países a través de la construcción de convenios entre entidades públicas y privadas para desarrollar nuevas tecnologías en el área de la salud como elemento fundamental para mejorar la calidad de vida de las poblaciones

## **1.3. Objetivos**

### **1.3.1. Objetivo General**

Desarrollar un sistema prototipo para la captura de movimientos de flexión/extensión, aducción/abducción del miembro inferior como interfaz de usuario para la interacción y monitoreo de actividad física.

### **1.3.2. Objetivos Específicos**

- Analizar y caracterizar la biomecánica, antropometría y anatomía funcional del miembro inferior y sus ejercicios asociados con flexión/extensión para el diseño mecatrónico del prototipo.

---

<sup>2</sup> [http://www.dane.gov.co/index.php?option=com\\_content&view=article&id=74&Itemid=120](http://www.dane.gov.co/index.php?option=com_content&view=article&id=74&Itemid=120)  
<sup>3</sup> [http://www.who.int/eht/eht\\_intro/en/index.html](http://www.who.int/eht/eht_intro/en/index.html), fecha de acceso

- Analizar y seleccionar los componentes mecatrónicos requeridos para diseñar el prototipo de captura de movimiento.
- Desarrollar e implementar el subsistema de procesamiento de datos para la consolidación y presentación de la información capturada.
- Diseño e implementación de un ambiente de realidad virtual que responda a los datos capturados para generar interés en la realización de los ejercicios a través de interacción e inmersión.
- Probar y validar el prototipo para realizar los ajustes correspondientes.

#### **1.4. Presentación del documento**

El presente documento se organiza en Capítulos de la siguiente forma: En el Capítulo 2 se presenta una revisión del estado del arte, donde, en primer lugar, se hace un repaso de cuales fueron primero videojuegos enfocados en el concepto de Exergame y en qué consistían, haciendo una referencia a la historia del Exergame, luego se echa un vistazo a que dispositivos están implementados hasta el momento con el enfoque que tiene el proyecto, y por último se revisa que tecnologías son la que actualmente se están utilizando en el área de los juegos dirigidos al sector salud.

En el Capítulo 3 se da a conocer el desarrollo que se lleva a cabo en el proyecto, comenzando por la documentación de la biomecánica y cinemática del miembro inferior, para luego describir cómo se va hacer la captura del movimiento del miembro inferior, que tecnología se va a utilizar y como se va implementar.

En el Capítulo 4 se presentan los resultados obtenidos, tanto del dispositivo como de la interfaz de usuario y la prueba de validación del prototipo que se llevan a cabo. Por ultimo en el Capítulo 5 se presentan conclusiones, que salen a partir del análisis de los resultados.

# CAPÍTULO 2: ESTADO DEL ARTE

El estudio, análisis, e validación de posturas se ha beneficiado con la evolución y desarrollo de sensores. Al inicio los métodos convencionales de captura de movimiento utilizaban actómetros (dispositivo para capturar aceleración y desaceleración), podómetros mecánicos, sensores piezo-cerámicos, posteriormente con la creación de acelerómetros integrados con otros sensores se abrieron amplias posibilidades como herramienta en aplicaciones médicas. A partir de estos avances se evidenció el potencial de las aplicaciones es el monitoreo de comportamientos, síntomas y cambios fisiológicos ya que estos tienen un gran impacto en el consumo de energía, presión arterial, patologías en la marcha y postura [5].

A medida que el hardware se volvió más portátil y las comunicaciones inalámbricas tomaron mayor fuerza, una nueva tendencia en la investigación y desarrollo de dispositivos se presentó alrededor del monitoreo de la salud humana. Con el objetivo de integrar varios dispositivos inalámbricos como sensores cinéticos, fisiológicos y ambientales, en [6] se propuso la implementación de una Wireless Body Area Network (WBAN) a través de la cual se implementa un aplicativo para rehabilitación física asistida por computador. Esta solución realiza el análisis en tiempo real de los datos obtenidos por los sensores, y los utiliza para realimentar al usuario generando avisos dependiendo del tipo de actividad y estado, así mismo envía la información para un servidor para posterior uso médico. Dentro de las ventajas ofrecidas por este tipo de metodología, se tiene como resultado que el sistema desarrollado es de bajo costo y poco invasivo para el monitoreo de actividades, también se obtiene de las pruebas realizadas la necesidad de mejorar aspectos relacionados con la configuración del sistema, personalización, integración y unificación.

La evolución de dispositivos para medir y analizar movimiento, abrió el camino para el desarrollo de múltiples herramientas entre las cuales se pueden encontrar aquellas basadas en sensores de tipo magnético, óptico, ultrasónico o mecánicos que requieren de equipo adicional de alto costo, trajes especiales ya sea para el uso de marcadores pasivos /activos o estructuras de tipo exoesqueleto, y adicionalmente grandes espacios para la captura de datos. Tomando en cuenta las características de algunos sistemas comerciales. En [7] se propone un sistema portátil para el análisis de marcha utilizando acelerómetros y giroscopios, consistiendo en el acoplamiento de los sensores en las articulaciones de cada miembro inferior para obtener información en el plano sagital. Para validar la información capturada los datos fueron comparados con otros obtenidos de un sistema óptico pasivo, resultando en un error bajo y probando el sistema viable para la análisis de marcha.

## 2.1. Historia del Exergame

Con la reciente tendencia en las empresas desarrolladoras de videojuegos de ofrecer experiencias con mayor interactividad e inmersión para aumentar el realismo, soluciones como el Kinect basado en procesamiento de imágenes para reconocimiento de movimientos [8], el Move<sup>4</sup> basado en marcadores ópticos activos con el mismo fin, y el Wiimote<sup>5</sup> utilizando giroscopios y acelerómetros, han adquirido fuerza en los mercados como elementos que integran varios sensores y procesamiento de señales embarcados en aparatos portátiles y de bajo costo comparados con el equipo de RV disponible en el mercado.

Como LaViola expresa en [9], los altos costos de los dispositivos tradicionales durante los últimos 20 años, los desarrollos se veían limitados solo a quienes poseían los medios y la financiación para investigar y realizar desarrollos en el área del 3D inmerso e interactivo. Recientemente, con los avances en electrónica y mecánica, en la actualidad se cuenta con dispositivos de bajo costo que han masificado los sistemas de visualización e interacción. Algunos ejemplos son el Wiimote, cuyo funcionamiento utiliza los movimientos del usuario para realizar las interacciones con un entorno virtual, el control realiza la captura de información utilizando un acelerómetro, un giroscopio y una cámara infrarroja. Algunas aplicaciones fuera del ámbito del entretenimiento permiten apoyar labores de teleoperación, rehabilitación o entrenamiento musical [9]. Este movimiento por parte de Nintendo inspiró a otros desarrolladores a innovar las UI, Sony y Microsoft lanzaron dispositivos que también responden a movimientos por parte del usuario, pero la mayor revolución la presentó Microsoft con su sensor Kinect donde a partir de imágenes capturadas con una cámara RGB, una cámara infrarroja y una de escala de grises, se consiguen generar mapas de profundidad que permiten la generación de esqueletos virtuales y reconstrucción tridimensional de espacios [10].

Conforme los dispositivos se masificaron, también el interés de curiosos, desarrolladores e investigadores en todo el mundo, permitiendo expandir los usos y aplicaciones de estos dispositivos en áreas no relacionadas con los videojuegos. Luego de ver el gran interés por parte de la comunidad y el gran desarrollo de SDK y API para el Wiimote, ninguno apoyado oficialmente por Nintendo, una vez lanzado el Kinect las API no oficiales y una por parte del fabricante del hardware se consiguen en la red. Siendo conscientes de la importancia e impacto del Kinect, Microsoft decidió lanzar su SDK oficial donde algunas de sus aplicaciones han permitido reconstruir ambientes reales [11][12], seguimiento de movimiento [13][14], navegación robótica [15].

En [9] RealDance toma como base poses para el reconocimiento de ciertos movimientos de baile. Para cada pose el usuario debe mantener una postura específica a partir de un tiempo  $t_0$  hasta un

---

<sup>4</sup><https://goqav.wordpress.com/category/future-technology/>

<sup>5</sup> <https://store.nintendo.com/ng3/browse/productDetailColorSizePicker.jsp?productId=prod150198>

$t_0 + N$ , donde N es un número específico de ritmo, de forma el éxito del juego se alcanza a través del porcentaje de intervalos imitados con suceso.

En [16], se utiliza el Wiimote para implementar un sistema de análisis y realimentación de gestos de un conductor de orquesta, quien a través de los movimientos de mano puede transmitir información como el tempo, ritmo, dinámica o expresiones durante la pieza musical. El Wiimote es usado debido a que permite realizar el seguimiento de movimientos 3D en la forma de posiciones y aceleraciones permitiendo una realimentación de los gestos. Para la adquisición de información se utilizaron varios Wiimote, segmentación de datos con base en la aceleración, tres modelos de análisis, y un método sonoro para el ritmo.

En [17], se toma como base el gran desarrollo que tienen hoy día los dispositivos no tripulados, a pesar de su autonomía aún requieren herramientas para la interacción con personas. Una forma de lograr una interacción adecuada entre robots y humanos, se basa en la forma como las personas se comunican entre sí, tomando como base los gestos de mano y brazo mapeados en conjuntos de comandos robóticos. El objetivo es mejorar la interacción del operador con la máquina y reducir los tiempos de entrenamiento. Como interfaz de entrada para la programación y uso del robot, se tomó el Wiimote con el cual se implementaron los siguientes sistemas de control: control como Joystick tradicional, movimiento de dirección a través del acelerómetro y control a través de gestos y brazos. Para la programación como Joystick, se utiliza el d-pad para indicar la dirección del vehículo, en el caso de los acelerómetros se utilizan las rotaciones en conjunto con los botones para definir la orientación del robot, y finalmente a través de gestos dinámicos se programan trayectorias a seguir.

## **2.2. Antecedentes de los juegos serios para la salud**

En [18], se implementa un videojuego basado en las interacciones posibles con el Wiimote. La finalidad es permitir la rehabilitación de muñeca a causa de derrames, fractura o pérdida de movimiento con la posibilidad de que los ejercicios puedan ser realizados desde casa. El sistema se compone del Wiimote funcionando como cámara, y un conjunto de LED IR posicionados a lo largo de la mano y antebrazo del usuario para monitorear sus movimientos durante la interacción.

Fuera de las aplicaciones documentadas que relacionan trabajos con el Wiimote, debido a los SDK disponibles para el sensor Kinect de Microsoft, desarrolladores e investigadores lo han utilizado para desarrollar diferentes herramientas. Dentro de las aplicaciones desarrolladas, se tienen programas para la captura de movimiento, que a través de características biométricas permite la identificación de usuarios. Este proceso determina la ubicación, tamaño del rostro y altura permitiendo la generación de esqueletos virtuales asociados para personalizar la experiencia de juego [19].



En el área médica se han realizado avances para el mejoramiento de rehabilitación motora al aprovechar la interacción natural que se puede alcanzar con el Kinect. Para medir la efectividad de la aplicación se realizaron medidas de ejercicios de flexión y extensión del brazo monitoreando los aciertos y efectividad de los movimientos realizados. El estudio de la conveniencia de esta herramienta en ejercicios del brazo permitió concluir que el Kinect es apropiado incluso para ser utilizado como herramienta en otros tipos de rehabilitación [20]. De forma similar, en robótica se han realizado algunas aplicaciones para la teleoperación de robots móviles a partir de gestos, algunos ejemplos en dispositivos terrestres para la detección de obstáculos, y otros en dispositivos aéreos para navegación e inspección [11]. En este mismo campo, los robots humanoides juegan un papel importante ya que se han realizado primeros avances para mapear movimientos humanos [21], que podrían resultar en la operación de exoesqueletos de una forma más natural.

Microsoft ha publicado en galería de desarrollos con Kinect<sup>6</sup>, una serie de aplicaciones desarrolladas como juegos serios con aplicaciones médicas como se presentan a continuación:

- Centro ayuda a niños autistas utilizando captura de cuerpo entero como complemento a las terapias de movimiento.
- Pacientes con accidente cerebrovascular utilizan la captura de movimiento como parte de sus terapias para que sus pacientes recobren movilidad.
- Como herramienta para navegación de imágenes médicas.
- Fisioterapia en casa con Kinect
- Tranquilizar niños pacientes a través de interactividad

Más allá de estas aplicaciones, Microsoft también documenta casos de éxito como el caso de la Universidad de Missouri donde utilizando este sensor para monitorear y medir cambios en el movimiento de personas de la tercera edad, permitiendo tomar acciones correctivas antes de que sea demasiado tarde. El sensor también está siendo utilizado como herramienta para la rehabilitación de pacientes con accidente cerebrovascular, fisioterapia en trabajadores industriales, niños con problemas de atención o hiperactividad entre muchas [22]. En el campo de la fisioterapia Microsoft reporta un caso de estudio sobre los efectos de realizar mediciones durante la realización de ejercicios como medio para optimizarlos, en este documento se analizan las ventajas y el potencial que dicha herramienta presenta en este campo [23].

### **2.3. Tecnologías enfocadas a la salud**

En la actualidad la masificación de dispositivos de interfaz de usuario está permitiendo que cada vez más usuarios los utilicen, esto representa una oportunidad de desarrollo de aplicativos que acerquen procedimientos o actividades relativas al ejercicio físico a los usuarios utilizando

---

<sup>6</sup> <http://www.microsoft.com/en-us/kinectforwindows/discover/gallery.aspx>

sistemas que pueden utilizar en cualquier lugar y que recopilar información para retroalimentación y posterior análisis por especialistas. Esta tendencia puede verse en el gran auge de los teléfonos inteligentes, consolas de videojuegos y televisores inteligentes, los cuales ofrecen interacción basada en gestos capturados a través de sensores electromecánicos, procesamiento de imágenes y de voz. También es posible ver la tendencia en desarrollar dispositivos que maximicen la experiencia basada en gestos y cuyo uso sea mínimamente invasivo, como lo propone la empresa Thalmic con su sensor de banda de brazo MYO<sup>7</sup> que utiliza señales miográficas y sensores de posicionamiento para que el usuario interactúe con un aplicativo sin necesidad de mouse o teclado; otra solución la propone la Universidad de Washington denominada WiSee<sup>8</sup> basada en el reconocimiento de gestos a partir de las interrupciones detectadas por un router inalámbrico, utilizando la red de área inalámbrica de la cual se dispone, superando la dificultad de sensores basados en línea de visión directa como el Kinect; SoundWave<sup>9</sup> de Microsoft busca capturar gestos a través del efecto Doppler captado por un micrófono cuando se realizan cerca del mismo como una alternativa a la captura de movimientos basada en imágenes; el LeapMotion<sup>10</sup> ofrece captura de gestos a través del reconocimiento de los dedos de ambas manos a través de cámaras infrarrojas; finalmente, otros dispositivos que también se están imponiendo y aun cuando no capturan información sobre movimiento corporal permiten la interacción con base en señales neurales como presenta Emotiv<sup>11</sup> con su NeuroHeadset.

Según Richard Bartle de la Universidad de Essex el concepto de Gamification se describe como “convertir algo que no es un juego en un juego” [58], los primeros usos de esta palabra provienen de los principios de los años 80 cuando por primera vez se comenzó a utilizar. La definición actual del termino planteada por Werbach en su libro “For the Win” es la siguiente “el uso de elementos de juego y técnicas del diseño de juegos en un contexto que este fuera de la definición de juego” [24].

Los juegos serios y la gamification han sido aplicados en distintos casos a pacientes en rehabilitación, uno de estos casos se presenta con personas que sufren de accidente cerebro-vascular como es el caso de Brick'a Break[25], donde el objetivo es romper bloques en la parte superior de la pantalla haciendo uso de una bola. En este caso la Realidad Aumentada (RA) permite que el paciente tome un pequeño bloque que posee un marcador en él, donde haciendo uso de este puede manipular el juego a través de sus manos, lo que mejora la fuerza de agarre además de reactivar las terminales nerviosas dañadas por el accidente cerebro-vascular, resultando en mejorías de los tiempos de respuesta además de conseguir resultados de forma más rápida que si hubiesen realizado una terapia convencional. Otro abordaje se tiene con pacientes que sufren de problemas lingüísticos, que son tratados mediante juegos de reconocimiento de

---

<sup>7</sup> <https://www.thalmic.com/myo/>

<sup>8</sup> <http://wisee.cs.washington.edu/>

<sup>9</sup> <http://research.microsoft.com/en-us/um/redmond/groups/cue/soundwave/>

<sup>10</sup> <https://www.leapmotion.com/>

<sup>11</sup> <http://www.emotiv.com/apps/epoc/299/>

signos y señas manuales [26], estos sirven como medio para mejorar los déficits de aprendizaje e interpretación de signos. Además se han adelantado tratamientos para personas con Parkinson por medio de juegos serios, que no sólo ejercitan los músculos de los pacientes sino que además mantienen el cerebro saludable por medio de ejercicios simples de asimilación y memoria (Assad, y otros, 2011). El tratamiento de adultos de la tercera edad también es uno de los casos más tratados dentro de los juegos serios, ya que permiten a las personas realizar los ejercicios en casa por medio del Kinect de Microsoft<sup>12</sup>, de esta forma no encuentran necesario tener que salir a la calle, las terapias practicadas desde la casa que incluyen taichí [27], juegos mentales [28], y ejercicios fisioterapéuticos para el fortalecimiento muscular [29] [30].

En conclusión entre las ventajas que ofrecen los aplicativos basados en juegos serios pueden considerarse las más relevantes:

- **Intensidad:** Los videojuegos serios permiten desarrollar los ejercicios planteados de forma en la que existe un desafío progresivo de fuerza para los usuarios, además de esto puede plantearse un itinerario de ejercicios que cumplan con la terapia requerida.
- **Repetición:** La actividad física consiste en la realización de ejercicios que trabajan las áreas afectadas por la lesión o enfermedad que se trabaja, normalmente una sesión consta de varias repeticiones de los mismos ejercicios, para eso es importante la facilidad de volver a realizar los mismos si es que falta algo por trabajar.
- **Motivación:** Al convertir la actividad física en un juego, los ejercicios dejan de ser molestos, ya que se brinda un objetivo al usuario que puede utilizar para mantener el interés por realizar la rehabilitación, eliminando también problemas de dolor, incomodidad y frustración por no poder realizar los ejercicios de forma correcta.

---

<sup>12</sup> <http://www.xbox.com/es-CO/Kinect>

# CAPÍTULO 3: DESARROLLO

En el presente Capítulo se presenta el desarrollo realizado para obtener el sistema de captura de movimiento, en primer lugar se analizará el miembro inferior a fin de identificar sus movimientos asociados debidos a su anatomía y biomecánica junto con los ejercicios para fortalecimiento. Seguido de esto, se realizará el diseño del sistema utilizando sensores inerciales y el desarrollo de un software para la realimentación del usuario dentro de un ambiente basado en juegos

## 3.1. Caracterización del miembro inferior y sus movimientos asociados

El cuerpo humano es un gran sistema complicado fisiológica, anatómica y estructuralmente. Debido a que el enfoque del proyecto es el diseño de un prototipo de captura de movimiento para el miembro inferior es necesario conocer de forma general el comportamiento del miembro para el cual el diseño está enfocado [31].

Un concepto básico indispensable es la planimetría que se va a utilizar, la cual se encarga de dividir el cuerpo por en secciones por medio de planos, como se presenta en la Figura 1.

La división de planos es la siguiente.

- Plano sagital: Divide en izquierda y derecha.
- Plano trasversal: Lo divide en inferior superior.
- Plano Frontal: Divide en anterior (Ventanal) y posterior (Dorsal).

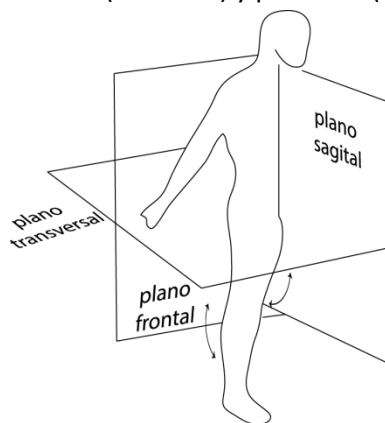


Figura 1 Planos de movimiento del cuerpo humano (tomado de: (Referencia))

Los movimientos de rehabilitación de flexión/extensión, aducción/abducción se encuentran principalmente en el plano sagital y en el plano frontal. En el caso de la caminata los movimientos

se concentran principalmente en el plano sagital y el movimiento principal de la cadera se encuentra en el plano frontal.

### 3.1.1. Miembro inferior

La estructura del miembro inferior humano está dividida en tres segmentos principales: la cadera, la pierna y el pie como se presenta en la Figura 2 Estrictamente hablando de la pierna humana es considerada como el segmento comprendido entre la cadera y el tobillo. Los músculos de la pierna están divididos en tres secciones: La sección superior y la sección inferior; Se mencionaran los principales músculos de la sección superior que influyen en los movimientos de flexión y extensión, como también aducción y abducción, como se presenta en la Tabla 1 (Referencia).

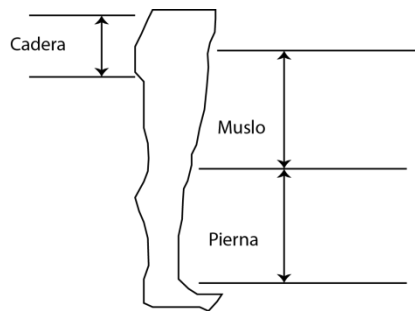


Figura 2 Sistema de soporte y actuación

Tabla 1 Grupos musculares

Nombre del músculo	Función
Aductores	Se encargan de acercar el muslo al centro del cuerpo
Tensor de la faja lata	Mover el muslo hacia fuera y flexionar la rodilla
Bíceps Femoral	Extender el muslo y flexionar la rodilla
Cuádriceps	Extender la rodilla y flexionar la cadera

### 3.1.2. Articulaciones de la pierna.

Las articulaciones son zonas de unión entre los huesos o entre los cartílagos del esqueleto y su función es permitir la movilidad de las diferentes extremidades del cuerpo humano. En el desarrollo de este proyecto el enfoque se hará en la articulación de la rodilla. La rodilla es la articulación más grande del cuerpo humano. Es una articulación compuesta, formada por la acción conjunta del fémur, tibia, rotula y dos discos fibrocartilagosos [31]. Los movimientos de la rodilla van guiados (y restringidos) por dos grupos de ligamentos:

- Ligamentos colaterales: medial y lateral.
- Ligamentos cruzados: anterior y posterior.

Las dos funciones básicas de la rodilla necesarias para este proyecto son la flexión y extensión. Los dos músculos principales que se encargan de estas dos funciones son:

- Flexión: Bíceps femoral
- Extensión: Cuádriceps femoral.

Es necesario conocer el rango de movimiento de la articulación para el desarrollo del dispositivo. Al flexionar la rodilla se genera un ángulo entre el fémur y la tibia, este ángulo varía dependiendo de la flexibilidad de las personas. Este ángulo va desde  $0^\circ$  cuando la pierna está totalmente extendida creando una línea recta el fémur con la tibia, hasta  $155^\circ$  cuando la pierna está totalmente flexionada, tocando el glúteo con el talón. Si se fuerza un poco el cuádriceps, se puede tener un hiperextensión de hasta  $-10^\circ$  tomando como referencia la pierna en posición recta en un ángulo  $0^\circ$  como se presenta en la Figura 3.

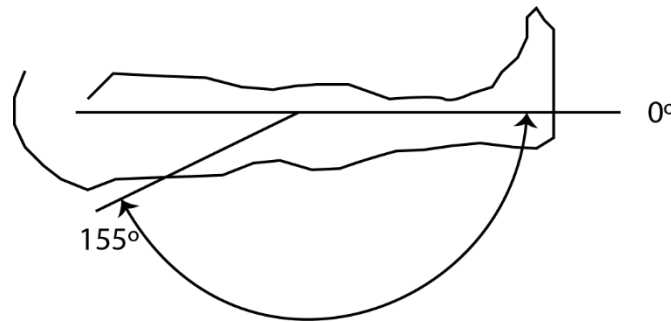


Figura 3 Amplitud de la flexión/extensión de la pierna

Como elemento de análisis la antropometría del miembro inferior permite conocer tamaño, proporciones y composición, aspectos que son de importancia en el diseño de las actividades de captura de movimiento debido a los espacios requeridos para la interacción del usuario con el exergame. Un estudio hecho en Colombia sobre los parámetros antropométricos de la población laboral colombiana<sup>13</sup> reporta que la longitud máxima, del miembro inferior, en el segmento medido desde el Trocánter mayor hasta el Cóndilo femoral lateral externo (muslo), es para el sexo femenino de y para 59.5 cm, el sexo masculino de 61.3, y para el segmento medido desde el Cóndilo femoral medial interno hasta Fascia plantar (pierna) es, para el sexo femenino 52.5cm y para el masculino de 56.6cm, obteniendo así, una altura máxima del miembro inferior para mujeres de 112cm y para hombres de 118.9cm

### 3.2. Biomecánica del miembro inferior - Cinemática

La cinemática estudia el movimiento de los cuerpos sin considerar las fuerzas o momentos que causan el movimiento, por otro lado en la robótica se refiere al estudio matemático para

<sup>13</sup> <http://es.slideshare.net/MriaCLemenciaAlzte/antropometria-jairo-estrada-percentiles-colombianos>

determinar la relación entre las posturas y velocidades de las articulaciones, contra la posición y velocidad de la herramienta o algún miembro terminal [32]

En este caso el análisis cinemático del miembro inferior se resuelve de forma directa. Primero se define el modelo, se analizan dos grados de libertad según las articulaciones de la cadera y rodilla respecto al movimiento de flexión/extensión como se presenta en la Figura 4.

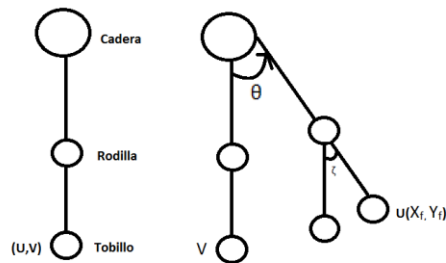


Figura 4 Modelo propuesto para simulación cinemática

### 3.2.1 Cinemática Directa

La cinemática directa describe de manera analítica las posiciones de cada articulación y el movimiento espacial de un punto final, a través de un análisis geométrico que se realiza por medio del método de Denavit Hartenberg [33]

Los parámetros propuestos para dos grados de libertad para los movimientos de flexión/extensión según la metodología de Denavit Hartenberg se presentan en la Tabla 2.

Tabla 2 Parámetros cinemática directa

Articulación	$\Theta_i$	$a_i$	$d_i$	$\alpha_i$
Cadera	$\theta$	0	$l_3$	0
Rodilla	$\zeta$	0	$l_4$	0

La multiplicación de las matrices de transformación de las articulaciones, es la matriz de transformación de todo el sistema, que se calcula a partir de los parámetros DH y se observa en la Ecuación 1.

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\alpha_i \cos\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \cos\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

### 3.3. Captura de Movimiento

Para la captura de movimiento debe de tenerse en cuenta el sistema de coordenadas por el cual el sistema debe de referenciarse, de esta manera se decide la posición que va a adoptar el sensor en el circuito y los ángulos que se van a medir con el mismo.

Teniendo el sistema de coordenadas establecido como se presenta en la Figura 5 a, luego se localiza el sensor en la mitad del muslo como se presenta en la Figura 5 b. A partir de este posicionamiento se configura la captura de información respecto a las variaciones angulares medidas por el acelerómetro y giroscopio.

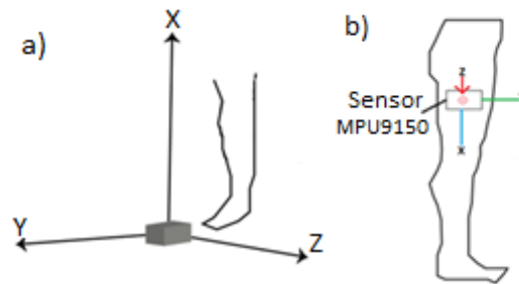


Figura 5 Ubicación del sensor

En la Figura 4 se presenta  $\theta$  como el ángulo de rotación de la cadera con respecto al eje X del sistema de coordenadas planteado anteriormente, para calcular la posición final del pie se aplica la transformada inversa de la matriz construida a partir de los parámetros de Denavit Hartenberg, teniendo en cuenta:

- La distancia  $d1$  debe ser dividida en la mitad de la longitud del muslo, ya que por comodidad el sensor va ubicado hacia la mitad del muslo.
- Los ángulos  $\theta$  y  $\zeta$  deben ser iguales, lo que quiere decir que el miembro inferior debe encontrarse en posición recta todo el tiempo, debido a que se debe disponer de otro sensor para obtener el ángulo  $\zeta$ , y así poder calcular la posición final U, teniendo en cuenta el movimiento de flexión y extensión de la articulación rodilla.

De esta manera obtenemos la posición final U que se ubica en el pie del jugador, si tener en cuenta el tamaño del pie. Por otro lado si hace la distancia  $d2$  igual 0, se obtiene la posición final de la articulación rodilla.



Después de obtener los datos se procede a configurar el módulo de transmisión inalámbrica Bluetooth y el puerto de la tarjeta de control, paso a seguir, desarrollar un programa que por medio de una interfaz visual permita evidenciar los datos de posición angular; asegurándose que los datos llegan de forma correcta, se procede a desarrollar la interfaz de usuario, la cual simula un juego, que debe permitir la interacción con el usuario y proveer la información obtenida al mismo.

### **3.4. Arquitectura del sistema.**

El sistema está compuesto de Hardware y Software orientado a capturar el movimiento y realimentar al usuario durante la realización del ejercicio con entorno virtual. A nivel de Hardware se cuenta con un sensor inercial compuesto de un acelerómetro y giroscopio a partir del cual se obtiene la posición, la velocidad y la aceleración del miembro inferior; estos datos son enviados a un microcontrolador Arduino Uno<sup>14</sup> por medio de un dispositivo Bluetooth que se encuentra contenido dentro de una carcasa que el usuario utiliza sujeto al muslo. La información se recibe por medio de Bluetooth en un computador que contiene el Software desarrollado en Processing<sup>15</sup> que recibe los datos, los procesa y define las acciones dentro del ambiente interactivo según los movimientos del jugador. El aplicativo permite guardar la información relativa a los movimientos del usuario para posteriormente realizar una retroalimentación basada e estos datos para que un especialista de salud acompañe y tome decisiones sobre el progreso del usuario.

Los componentes que hacen parte del diseño de esta arquitectura se describen a continuación:

- Usuario: Es la persona quien se va beneficiar de los objetivos del desarrollo del dispositivo, este tendrá la oportunidad de interactuar directamente con el juego por medio del dispositivo de comunicación y a su vez se va favorecer al mejorar su salud.
- Transductor: Es el componente que convierte energía o información desde un sistema en energía o información para otro sistema [34]. En este caso es el acelerómetro el cual convierte la aceleración a la que es sometido en información para poder cuantificarla.
- Juego<sup>16</sup>: Hacer algo con alegría y con el solo fin de entretenerse o divertirse. El juego desarrollado tiene como objetivo encestar una pelota en una canasta con el movimiento de la pierna.
- Puntaje: Tiene como objetivo motivar al usuario Es la ganancia o recompensa que va a recibir el jugador, la cual está sujeta a que el movimiento de su pierna sea realizado correctamente lo que en el juego sugiere que la pelota va a ser encestanda.
- Datos: Son los datos provenientes de las lecturas de la aceleración del sensor, estos datos son procesados y luego comparados con datos de una curva ideal del movimiento a

---

<sup>14</sup> <https://www.arduino.cc/en/Main/arduinoBoardUno>

<sup>15</sup> <https://processing.org/>

<sup>16</sup> <http://lema.rae.es/drae/srv/search?id=zOcyMLMfQDXX2B6Qr4ue>

realizar, dependiendo de error entre la curva del movimiento realizado y la curva del movimiento ideal se determina si el jugador cumple el objetivo, en este caso encestar la pelota.

- Gráficas del movimiento: Son la gráficas que se presentan al jugador después de ser tomada una decisión resultante de la comparación de datos.
- Informe de movimientos realizados y puntajes.

La arquitectura de sistema propuesta basada en los elementos presentados se puede observar en la Figura 6.

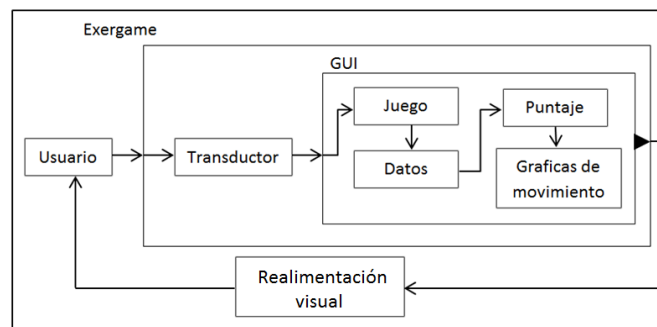


Figura 6 Arquitectura del sistema propuesto

### 3.5. Implementación de la captura de movimiento y transmisión de datos

En esta sección se presenta la implementación del sistema de captura de movimiento tomando como base las especificaciones técnicas de los componentes y los movimientos del usuario.

#### 3.5.1. Implementación de captura de movimiento.

El movimiento de flexión y extensión que se captura será alrededor del eje Z del sistema de coordenadas propuesto anteriormente y la abducción y aducción en el eje alrededor del eje Y. El sistema coordinado se localiza en la cadera y a partir de este se traslada el sensor hacia la mitad del muslo donde se posiciona debido al uso de un accesorio para portarlo de forma cómoda como se presenta en la Figura 21. Sin embargo, debido a que el sensor estará localizado a lo largo del muslo, la rotación siempre va a ser la misma como se presenta en Figura 7 Para calcular la posición de la rodilla es necesario que el usuario provea la información de la longitud del muslo y de la pierna.

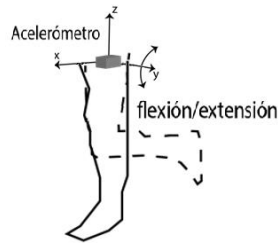


Figura 7 Grados de Libertad del sensor asociado a los planos de movimiento

La captura de movimiento se realizó a través del sensor inercial MPU9150<sup>17</sup> SparkFun, este se escogió porque permite integrar información del acelerómetro y del giroscopio haciendo una fusión de datos, que tiene como ventajas; reducir la incertidumbre e incrementar la precisión de los valores censados, aumentar la confiabilidad en caso de que uno de los sensores falle, y recolectar información complementaria del entorno que se está caracterizando, lo cual no se puede obtener con un solo sensor; todo esto con el objetivo de obtener un registro datos más adecuado. La transmisión de datos se realiza a través del protocolo I2C al microcontrolador, como se presenta en la **¡Error! No se encuentra el origen de la referencia.**

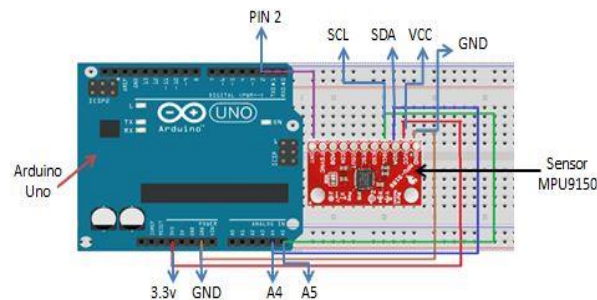


Figura 8 Circuito de conexión MPU9150

Los pines para la disposición del protocolo I2C en la placa Arduino Uno son A4 y A5 que corresponden al SDA (Línea de datos) y SCL (Línea de reloj) respectivamente; en el sensor MPU9051 de Sparkfun se encuentran en el tercer y cuarto pin en el mismo orden. La manera correcta de conectar los dos dispositivos es: polarizar el MPU9051 de Sparkfun con 3.3 v, el voltaje ya viene regulado del Arduino, así que no hay problema en conectar directamente, unir las tierras, conectar el SDA del Arduino con el SDA del sensor, así mismo el SCL del sensor con el SCL del Arduino, por último el programa está hecho para que cada vez que el sensor detecta una señal de interrupción envíe el dato a través del SDA, por tanto es necesario conectar el pin INT del sensor, que es el último, al pin 2 del Arduino, que es el que está dispuesto en el Arduino para enviar la señal.

<sup>17</sup> <https://www.sparkfun.com/products/11486>

Para realizar la captura de información se tomó como referencia la librería disponible junto con la documentación<sup>18</sup> del sensor.

En el programa se hace uso de tres librerías, la primera, llamada "I2Cdev", se utiliza para acceder a funciones que permiten conectar fácilmente módulos I2C de otros dispositivos con la placa Arduino; la segunda, "MPU6050\_6Axis\_MotionApps20.h", la cual permite acceder a las funciones que a partir de datos del sensor devuelven datos de magnitudes físicas como los son la velocidad, la aceleración, la gravedad o el ángulo en cualquiera de los tres ejes; y por último es necesario incluir la biblioteca "Wire" para configurar el módulo I2C del Arduino.

El programa está dividido en tres partes, en la primera se incluyen bibliotecas, se realiza la declaración de variables y declaración y definición de funciones; en la segunda parte se observa en la función "Setup" donde se configuran los diferentes módulos que se van a utilizar como lo son el módulo de transmisión serial y el módulo I2C; y en la tercera se realiza el desarrollo del programa. El diagrama de flujo que lo representa se da a conocer en la Figura 9

En los sensores MARG (Magnéticos, Velocidad angular, Gravedad), cada uno de los sensores de los cuales están compuestos presenta algunas ventajas y desventaja en las mediciones; como errores de deriva, sensibilidad al ruido mecánico o a inferencias magnéticas. El objetivo de los algoritmos de fusión de datos es, reducir el déficit de un sensor, con la información complementaria de los otros sensores, proporcionado información más precisa sobre la orientación del dispositivo.

El acelerómetro y el magnetómetro nos dan coordenadas de un vector en el sistema del sensor, ya que conocemos la orientación de estos vectores en el sistema tierra, que son, gravedad y campo magnético terrestre, se puede calcular la orientación del sistema del sensor con respecto al sistema de la tierra, con el inconveniente de que existen infinitas orientaciones que puede tener el sensor, con coordenadas iguales a las del vector, que son las rotaciones del sistema con el vector como eje de giro, por lo cual se implementa el algoritmo de Madgwick [36].

Este algoritmo se encarga de representar la rotación del sistema por medio de cuaterniones, estima la orientación del sistema en base a los vectores de gravedad y campo magnético y luego pondera las tres estimaciones obtenidas para obtener la solución final. Para evadir el problema de las soluciones infinita, utiliza el método del gradiente descendiente tomando la rotación más pequeña entre el sistema sensor, con respecto al sistema tierra [36]. El algoritmo se implementa en la función *MadgwickQuaternionUpdate*<sup>19</sup>

---

<sup>18</sup> [https://github.com/sparkfun/MPU-9150\\_Breakout/tree/master/firmware](https://github.com/sparkfun/MPU-9150_Breakout/tree/master/firmware)

<sup>19</sup> <https://github.com/kriswiner/MPU-9150/blob/master/quaternionFilters.ino>

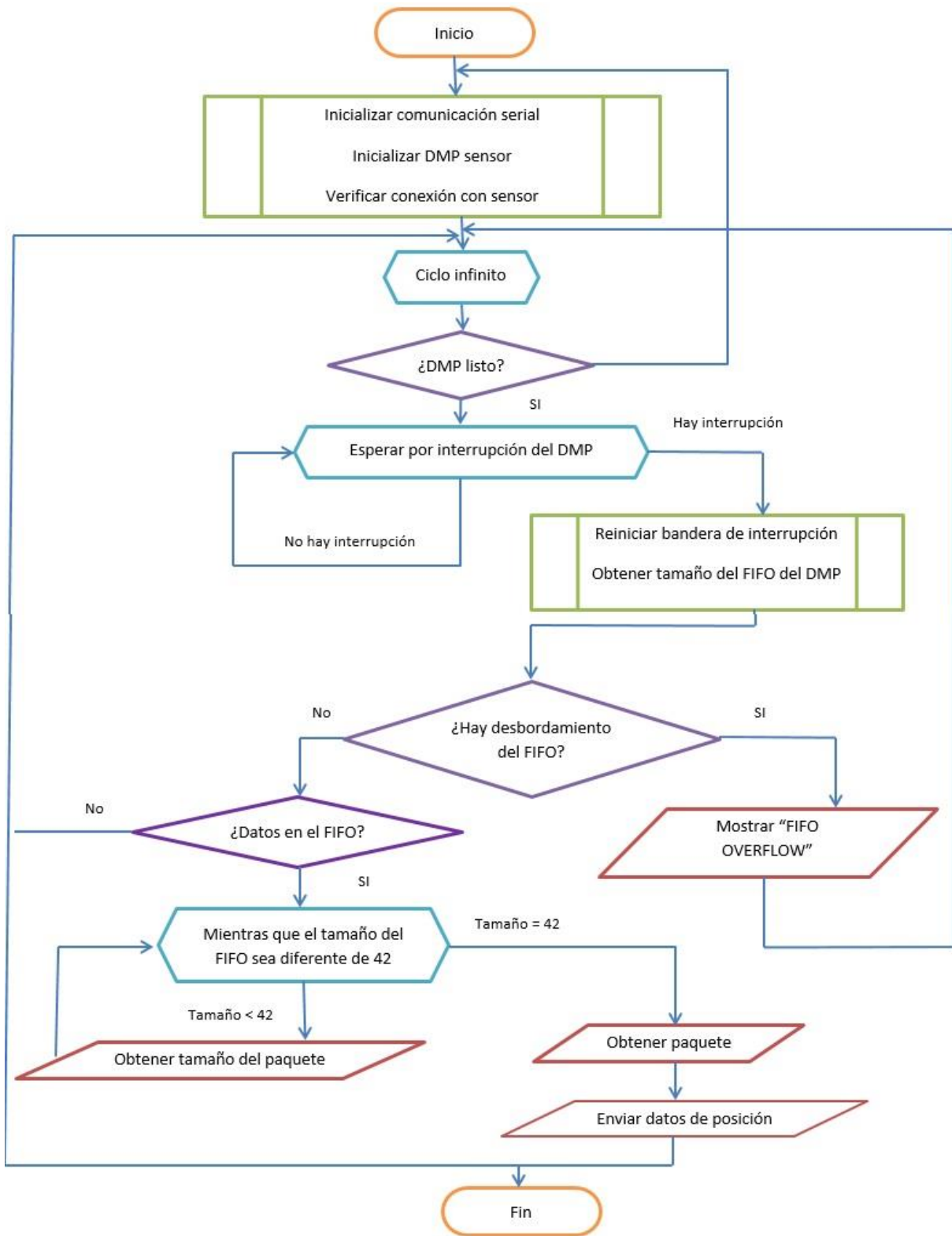


Figura 9 Diagrama de flujo, programa de conexión con MPU9150.

En la Figura 10 se presenta un diagrama donde se presta el algoritmo planteado en el algoritmo para fusión de datos de los sensores inerciales, donde se observa cómo se obtienen las rotaciones durante la flexión y extensión del miembro inferior.

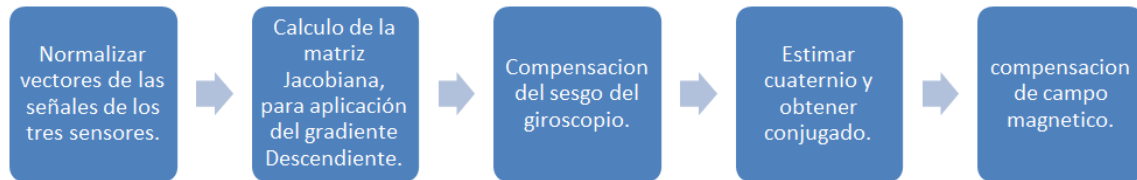


Figura 10 Algoritmo de Madgwick

A continuación en la Figura 11, se muestra una captura de pantalla donde se evidencian los datos del sistema de sensores inerciales después de ser implementado

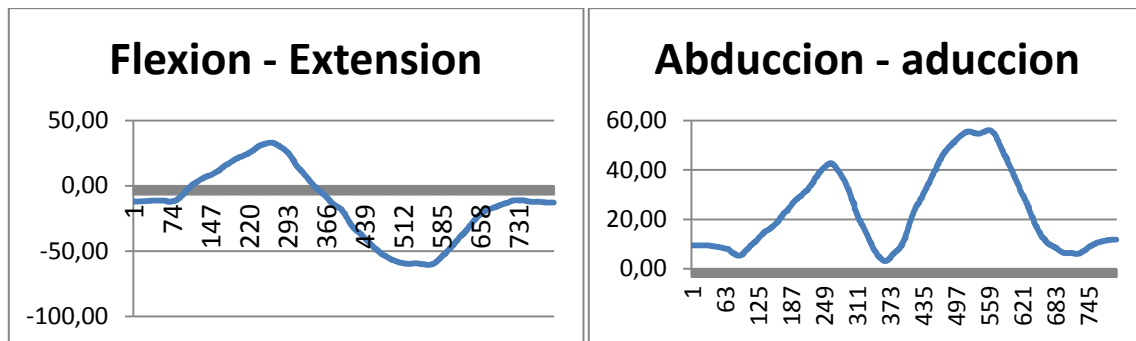


Figura 11 Datos obtenidos del sensor, movimientos: Flexión – Extensión, Aducción – Abducción.

### 3.5.2. Implementación de transmisión de datos.

Antes de incluir el módulo Bluetooth en el dispositivo, este se debe configurar a partir de los comandos AT. Un comando AT, se utiliza para configurar o indicar una acción a un módem por medio de instrucciones que comienzan con las letras AT, que hacen la abreviación de (ATtention), cada modem debe tener una lista de los comandos AT que soporta en su hoja de especificaciones [36]. Para este procedimiento, hay que hacer un circuito temporal, el cual se muestra en la Figura 12.

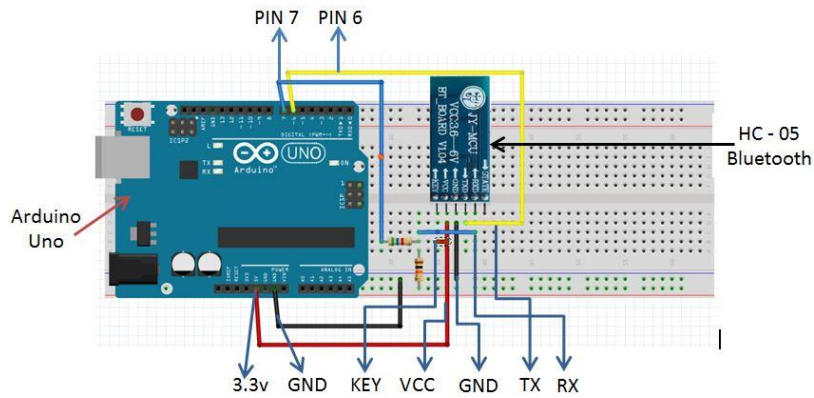


Figura 12 Circuito de conexión con el modulo Bluetooth.

Los comandos AT que se utilizan para modificar la configuración del módulo Bluetooth, según las necesidades del proyecto se indican en la Tabla 3:

Tabla 3 Comandos AT modulo Bluetooth HC-05

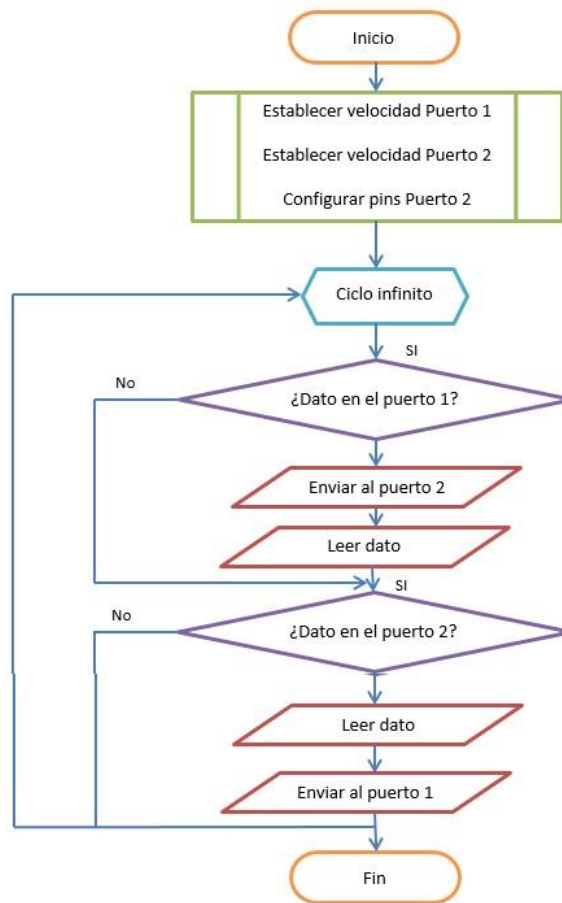
Comando AT	Respuesta	Parámetro	Propósito
AT	OK	NONE	Comprobar conexión
AT+ORGL	OK	NONE	Retornar a los valores por defecto
AT+VERSION?	+VERSION<Parámetro> OK	Numero de versión	Preguntar por el número de versión del dispositivo
AT+NAME=<Parámetro>	OK	Nombre	Establecer nombre del dispositivo.
AT+NAME?	+NAME<Parámetro> OK	NONE	Preguntar por el nombre del dispositivo.
AT+ROLE	OK	0 => Esclavo 1 => Maestro 2 => Ciclo Esclavo	Establecer el Rol que va tomar el dispositivo.
AT+ROLE?	+ROLE<Parámetro> OK	NONE	Preguntar el Rol actual del dispositivo.
AT+PSWD=<Parámetro>	OK	Clave	Establecer la clave
AT+PSWD?	+PSWD : <Parámetro>	NONE	Preguntar por la clave actual.

AT+UART=<Parámetro1>,<Parámetro2>,<Parámetro3>	OK	Parámetro1: 4800,9600,19200,38400 57600,115200,234000 460800,921600,1382400 Parámetro2: 0 => 1 Bit,1 => 2 Bits Parámetro 3: 0 => Ninguno,1 => Impar,2 <= Par	Establecer los parámetros de transmisión serial; La velocidad de transmisión, Bit de parada y Bit de paridad respectivamente.
AT+UART?	UART=<Parámetro1>,<Parámetro2>,<Parámetro3> OK	Ninguno	Preguntar por los parámetros de transmisión serial.
AT+POLAR=<Parámetro1>,<Parámetro2>	OK	Parámetro1: 0 => PI08 salida en nivel bajo 1 => PI08 salida en nivel alto Parámetro2: 0 => PI09 salida en nivel bajo 1 => PI09 salida en nivel alto	Establecer el nivel de voltaje de salida del PIN y el LED que indica el estado de conexión del dispositivo.
AT+POLAR?	+POLAR=<Parámetro1>,<Parámetro2> OK	Ninguno	Pregunta por los parámetros que establecen el nivel de voltaje de salida del PIN y el LED que indica el estado de conexión del dispositivo.

Para un correcto funcionamiento el dispositivo Bluetooth debe configurarse de la siguiente forma: Nombre: HC-05, Rol: Esclavo, Clave: "1234", Velocidad de transmisión: 115200 bit/s, Bit de parada: Uno, Paridad: Ninguna, Indicador de estado: LED encendido, salida en estado alto

El código que se utiliza para configurar los comandos AT del módulo Bluetooth se adjunta en el Apéndice 2 a su vez presentado en la Figura 13. En el programa se hace uso de la librería "SoftwareSerial" para utilizar otro puerto de comunicación serial con el que se transmite los datos al dispositivo Bluetooth y los datos de respuesta se visualizan en el monitor serial del software de Arduino. Este puerto se configura a una velocidad de transmisión de 38400 ya que es la velocidad que tiene por defecto el dispositivo Bluetooth HC-05.





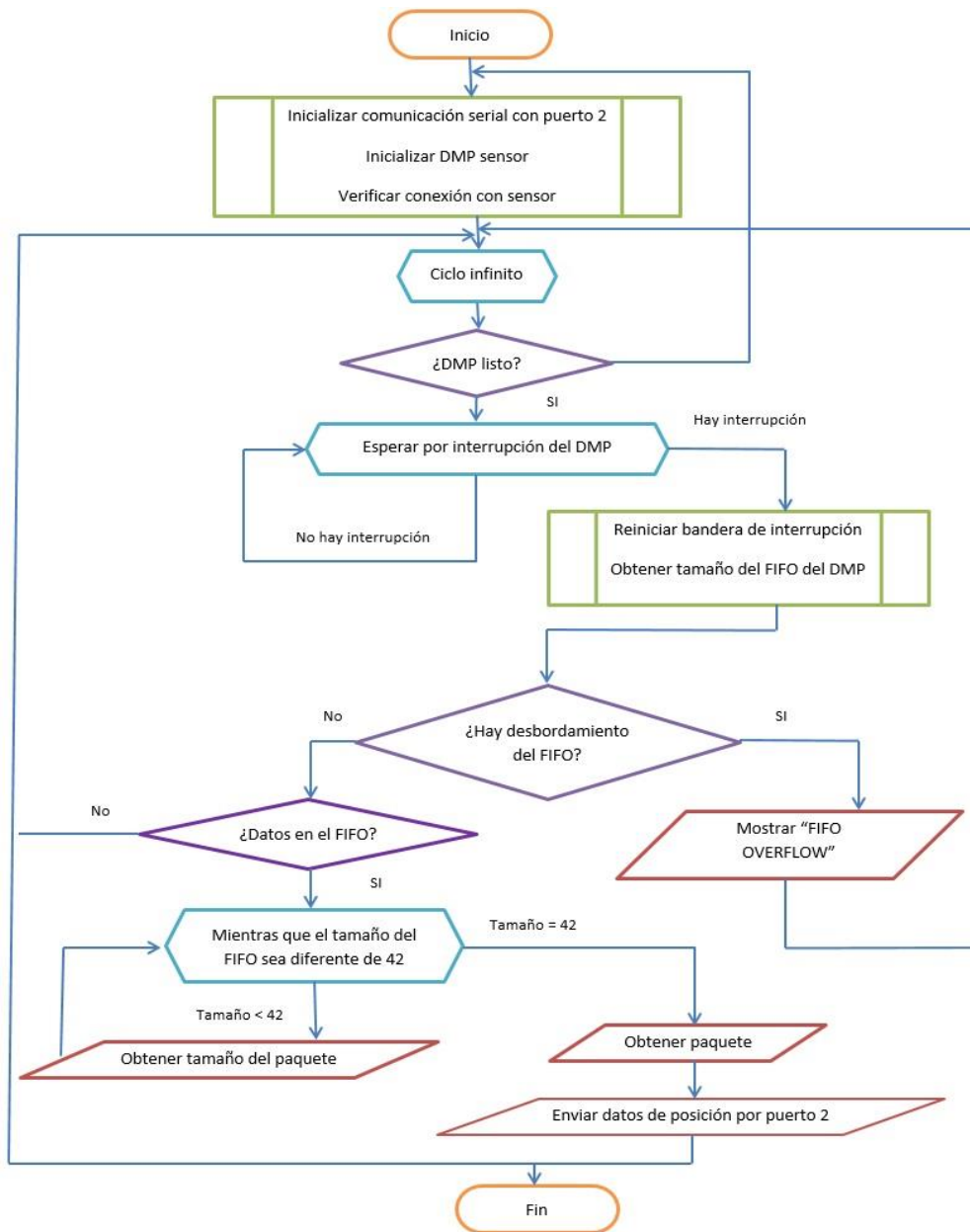
**Figura 13 Diagrama de flujo, Programa de configuración con comandos AT.**

En la Figura 14 se presenta la respuesta de los comandos AT en el Monitor Serial del Software de Arduino obtenida de la comunicación con el sensor siguiendo el diagrama de la Figura 14.

```

Inicio de comandos AT:
OK
+VERSION:2.0-20100601
OK
+NAME:HC-05
OK
+ROLE:0
OK
+PSWD:1234
OK
+UART:115200,0,0
OK
+POLAR:1,1
OK
  
```

**Figura 14 Monitor Serial, Configuración módulo Bluetooth.**



**Figura 15 Diagrama de flujo, Programa conexión a sensor MPU9150 y sensor Bluetooth.**

La librería, "SoftwareSerial", permite configurar los pines en los que se quiere realizar la transmisión serial. En este caso se configuran los pines 6 y 7 para realizar la transmisión serial, RX y TX respectivamente; se crea un objeto para poder acceder a ellos y además poder configurar la velocidad de transmisión, la cual se escoge de 115200 baudios para lograr un monitoreo fluido.

Se procede a hacer un programa en el Software de código libre Processing para visualizar los datos que se transmiten desde el dispositivo de monitoreo, el código se adjunta en el Apéndice 4, el diagrama de flujo se muestra, en la Figura 16 a continuación.

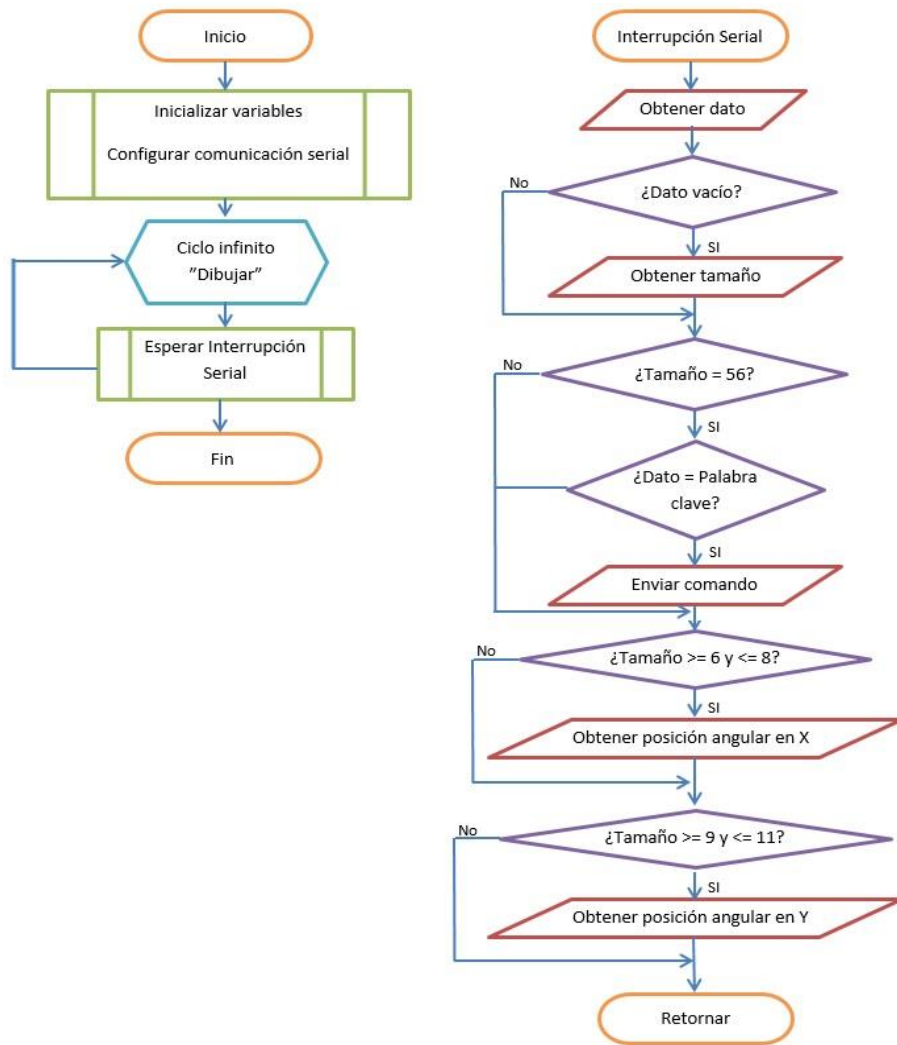


Figura 16 Diagrama de flujo, Programa de recepción de datos en Processing.

En la Figura 17 se evidencian datos de posición transmitidos por el módulo Bluetooth, resultado del programa anteriormente presentado. Donde el ángulo Pitch, representa la rotación alrededor del eje Y, lo cual expresa la posición angular del movimiento de aducción/abducción de la cadera; El Ángulo Yaw, representa la rotación alrededor del eje Z, que expresa la posición angular del movimiento de flexión/extensión de la cadera.

```
Done Saving.  
pitch  
-8.08  
yaw  
81.294174  
pitch  
-8.07  
yaw  
81.294136  
pitch  
-8.07  
yaw  
81.29865  
pitch  
-8.07  
yaw  
81.29861  
1 - 65
```

Figura 17 Processing 2.2.1, Datos de posición transmitidos.

### 3.6. Diseño del Aplicativo

La interfaz de usuario se diseña conforme el objetivo del aplicativo que es tomar la información de la captura de movimiento como método de interacción en el exergame. Para esto se definen los siguientes elementos del exergame:

#### 3.6.1. Objetivo del juego

Realizar movimientos de flexión y extensión de muslo con la finalidad de hacer el mayor puntaje a partir de los movimientos correctamente realizados. El juego propone, seguir el movimiento ideal presentado en la gráfica de la interfaz, con el movimiento de flexión/extensión y abducción/aducción realizado con el miembro inferior, con la finalidad de patear un balón, el cual seguirá una trayectoria que depende de la coincidencia del movimiento del jugador con el movimiento de la gráfica ideal, de esta forma, entre las gráficas sean más parecidas, el balón caerá más cerca de su objetivo, que son dos canastas que representan el éxito del movimiento, como también la cantidad de puntaje o reconocimiento que el jugador merece.

#### Reglas

El juego consta en realizar un movimiento con el miembro inferior que imita el pateo de un balón, durante el transcurso del movimiento se medirán la velocidad, aceleración y posición de la pierna para calcular la trayectoria que va a seguir el balón, éste debe ser encestado en una cesta que está dispuesta a una determinada distancia del personaje que representa el jugador. Si el balón es encestado aumentara el puntaje del jugador, y si el movimiento del jugador sigue una trayectoria ideal, se recompensa con unos puntos de más.

En la interfaz del juego además de la gráfica de movimiento que se presenta durante todo el tiempo en el ambiente de realidad virtual y donde el jugador puede observar el movimiento que

debe seguir, la opción de reporte, guarda y genera las últimas dos ejecuciones del jugador, espacio en el que se puede apreciar la calidad de la ejecución en el movimiento por parte del usuario, adicional a esto el programa también genera un documento en archivo de texto donde guarda los datos muestreados de las ejecuciones, con la intención de un análisis posterior.

### **3.6.2. Ambiente y personajes**

En el ambiente del juego se involucra cuatro botones diferentes donde el jugador puede elegir entre: Jugar, estadísticas, puntaje e información. Por otro lado se encuentra una gráfica, en la cual el jugador puede observar el movimiento que está realizando con el miembro inferior en tiempo real.

El juego involucra un único personaje que es aquel que representa el jugador y el movimiento que realiza con el miembro inferior a evaluar.

### **3.6.3. Modelo y Animación**

Para el desarrollo de la modelación y animación en este proyecto se escogió el programa de software libre Processing. En cuanto a la modelación, todo el ambiente y el personaje del juego se realiza a través de figuras primarias como rectángulos, triángulos, círculos etc. Por otro lado la animación se hace con código consolidado en funciones que permite hacer la traslación y rotación de las imágenes. El escenario fue escogido inicialmente para hacerlo mucho más sencillo, intuitivo y agradable al usuario, el avatar es el objeto principal en el ambiente de realidad virtual, pues es por medio de este que el jugador se va a sentir identificado, se pensó en un juego de balón, porque un balón es un objeto que desde hace muchos años tiene gran acogida en las personas y para hacerlo divertido se involucran dos canastas para acumular puntaje.

### **3.6.4. Prototipado del Juego**

En el prototipado del juego se inició con bosquejos en la interfaz muy básicos, desde donde inicialmente solo se reconocía el movimiento del miembro inferior, posteriormente se fueron adicionando poco a poco las demás características al ambiente de realidad virtual, tal como un avatar más completo, graficas de movimiento, reportes, se incluyó el segundo movimiento (aducción-abducción) entre otros. En la Figura 18, se puede observar el proceso de la evolución de los primeros ambientes de realidad.

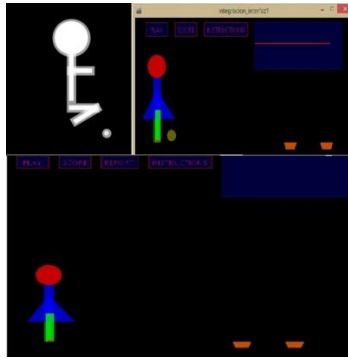


Figura 18 Primeros ambientes

En Figura 21 se puede observar la interfaz completamente terminada.

### 3.6.5. Motor del juego

Para el desarrollo del exergame se utilizó Processing, software que permite llamar funciones de interrupción orientadas a periféricos como teclado, mouse, puerto serial etc. Por medio de estas funciones se hace la integración del juego con el dispositivo de monitoreo como también con el usuario. En un principio se tenían en mente varios motores para desarrollar el juego, finalmente se seleccionó Processing porque además de ser un software libre, como principal característica, la plataforma de Arduino es derivada de Processing, lo que permite una mejor emparejamiento de datos entre el dispositivo y la interfaz. La plataforma processing está desarrollado principalmente como software de manejo y procesamiento de imágenes, por lo que se puede desarrollar una interfaz casi sin ningún problema.

Se plantean dos funciones de interrupción por periféricos; La interrupción por transmisión serial, genera un evento cuando el puerto serial declarado en el programa, recibe un dato. Cuando este evento se genera, se procede a leer los datos transmitidos por el módulo Bluetooth de la posición angular del miembro inferior; La otra función produce un evento cada vez que el mouse en el computador ejecuta un click, cuando se genera este evento, se evalúa el lugar en el espacio 2D en el que fue ocasionado, de esta maneras se determina que botón fue oprimido en la interfaz;

### 3.6.6. Física

Ya que la meta del juego es que el jugador consiga realizar el movimiento a partir del golpe a pelotas hacia canastas que le provean puntajes según los aciertos, la física en el juego está directamente relacionada con el movimiento parabólico que sigue la pelota, que describe la ecuación 2 y 3. Para conseguir una representación adecuada del movimiento parabólico se calcula la velocidad se como variable dependiente de la magnitud del error promedio, que resulta de la comparación de la gráfica que se obtiene del movimiento que se está ejecutando y la gráfica que tiene el movimiento ideal que se debería llevarse a cabo.

$$X = x_0 + v_0 t \quad (2)$$

$$Y = y_0 t - v_0 t - 0.5gt^2 \quad (2)$$

$$Y = y_{max} t + v_0 t + 0.5gt^2 \quad (3)$$

Dónde,

- X, es la posición Final de la bola de forma horizontal.
- Y, es la posición final de la bola en forma vertical.
- Y, es la coordenada inicial de la bola en Y.
- X, es la coordenada inicial de la bola en X
- V, es la velocidad inicial de la bola
- g, es la gravedad predeterminada, y
- t, es el tiempo, que este caso, cada unidad aumenta al ejecutarse la función de dibujar.

### 3.6.7. Programación

La programación de la tarjeta arduino con sus componentes se realizó de la siguiente forma: Declaración de variables, en variables de interfaz dinámica, variables de interfaz estática y variables que atienden la transmisión serial y los eventos del mouse; se configuran las dimensiones del juego, se inicializan variables y los protocolos de transmisión serial e I2C; se renderiza la interfaz de usuario, ambiente, modelos y animación; y por último la declaración y definición de funciones de interfaz estática, dinámica y funciones que atienden interrupciones de periféricos. En la Figura 19, se presenta un diagrama de flujo alusivo al código que se desarrolló para la realizar la interfaz de usuario, que hace la simulación de un juego; El código se adjunta en al Apéndice 5.

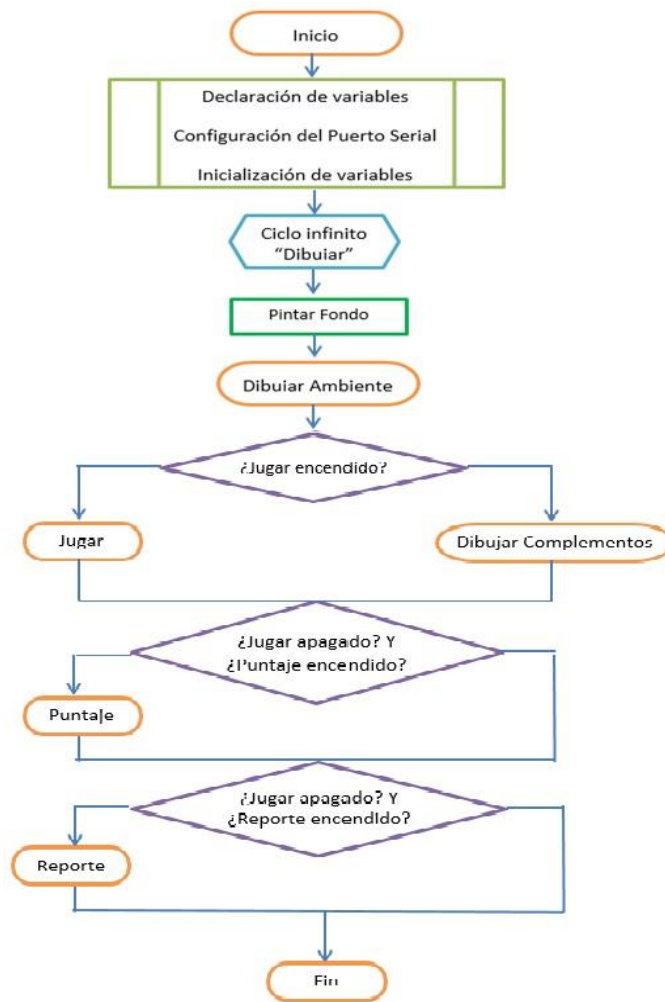


Figura 19 Diagrama de flujo, Interfaz-Juego.



# CAPÍTULO 4: RESULTADOS

En este capítulo se presentan los resultados obtenidos del desarrollo del proyecto, imágenes del prototipo del dispositivo de monitoreo, de la interfaz de usuario. También el desarrollo de un instrumento para realizar pruebas de validación del prototipo.

## 4.1 Dispositivo obtenido

El dispositivo que se obtuvo consta de una placa Arduino Uno conectada por medio de un Protoboard a un sensor inercial MPU9150 de Sparkfun y a un módulo Bluetooth HC-05. El dispositivo es desarmable, de esta forma se permite realizar algún arreglo, mantenimiento o un futuro desarrollo. En la Figura 20 se da a conocer una fotografía del dispositivo final para este proyecto.



Figura 20 Dispositivo obtenido.

El dispositivo se ubica en la parte lateral de la pierna entre la cadera y la rodilla de acuerdo al eje de coordenadas que se plantea en el capítulo anterior. En la siguiente Figura 21, se muestra una fotografía en la que se observa la ubicación del dispositivo y el accesorio que se utiliza para sujetarlo al miembro inferior.



Figura 21 Ubicación del dispositivo.

Con el accesorio como se muestra en la Figura 21 presenta la posibilidad de que el dispositivo resbale con los diferentes materiales de ropa, que este usando el jugado; por este motivo se toma la decisión de cambiar el accesorio que sujeta el dispositivo por el que se presenta en la Figura 22 el cual cuenta con dos sistemas de sujeción, los cuales son ajustables, uno en la cadera y otro sobre el muslo del miembro inferior, de esta forma se evita que el dispositivo resbale.



Figura 22 Nuevo accesorio

El dispositivo se energiza con un arreglo de cuatro pilas doble A que proporcionan la fuente de alimentación de un total de 6 voltios, con la posibilidad de aportar 350mA/hora.

## 4.2 Interfaz de usuario

En la Figura 23, se muestra una imagen de la interfaz de usuario final, desarrollada en el IDE Processing 2.2.1.

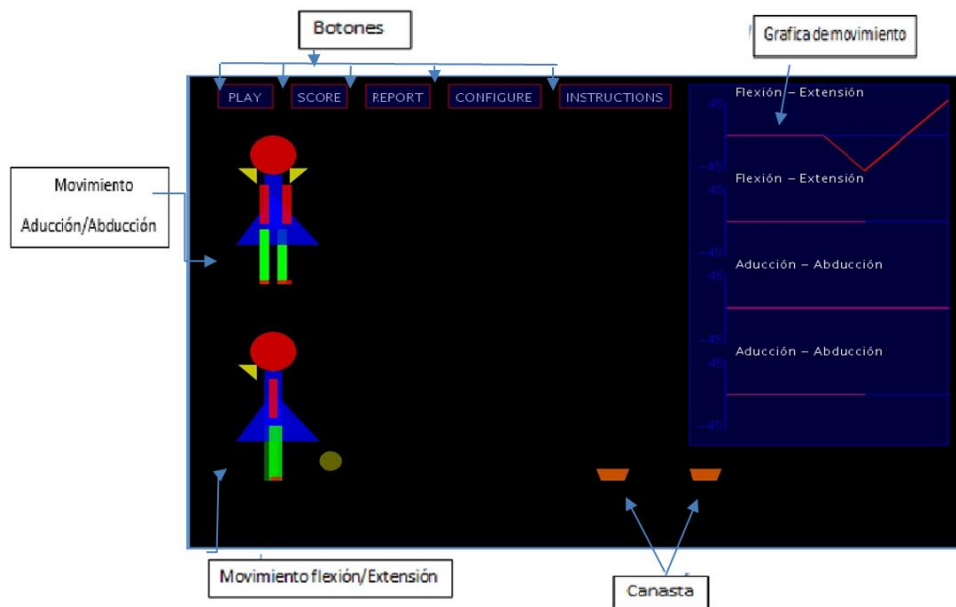


Figura 23 Interfaz de usuario en Processing

En la interfaz se visualizan cinco botones diferentes, su funcionalidad se explica a continuación:

- Opción jugar: en esta opción el juego permite ver una gráfica del movimiento que está realizando el jugador en tiempo real, por medio de esta grafica se pretende que el jugador sea consciente de que tan parecido sea su movimiento a la ejecución del movimiento ideal. A demás permite visualizar al jugador el movimiento del miembro inferior a través del personaje que lo representa; por otro lado se observa la animación del balón dirigiéndose a la cesta, si fue exitosa la cesta o no. La anterior información se puede visualizar en la Figura 24.
- Opción Puntaje: Esta opción permite al jugador visualizar el puntaje que lleva sumado desde que comenzó a jugar. Es importante aclarar que el juego no tiene una base de datos donde quede registrado el puntaje, por tanto, cada vez que se reinicie la interfaz el puntaje también lo hará.
- Opción Reporte: En esta opción se presenta al jugador los últimos cuatro movimientos y el movimiento ideal que este debe hacer, para que él jugador mismo pueda comparar.

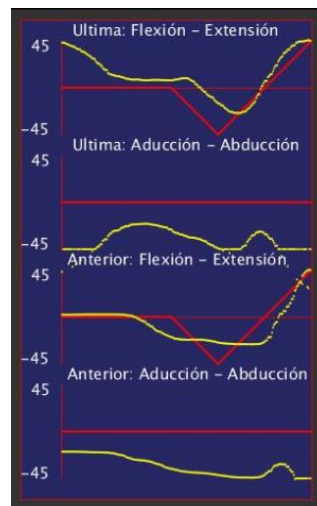


Figura 24 Opción reporte.

Donde,

- Grafica del movimiento realizado por el jugado.
- Grafica del movimiento ideal.

### 4.3 Validación

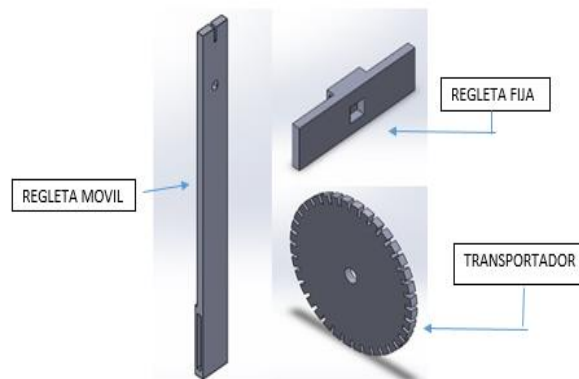
Se desarrolló una herramienta complementaria con el objetivo de comparar y validar los datos de las rotaciones obtenidas por parte del acelerómetro y giroscopio. El dispositivo rotacional está inspirado en una herramienta que se usa comúnmente para medir la amplitud rotacional de las

articulaciones denominada Goniómetro. El objetivo de la herramienta consiste en medir la situación de un segmento corporal con relación a otro, separado del primero por la articulación estudiada, o con relación a un elemento de referencia constante [37] El goniómetro de dos ramas está compuesto de una rama fija, que contiene la escala angular, y una móvil la cual se coloca sobre el segmento móvil de la articulación a evaluar, como se presenta en la siguiente .



**Figura 23 Goniómetro convencional de dos ramas**<sup>20</sup>

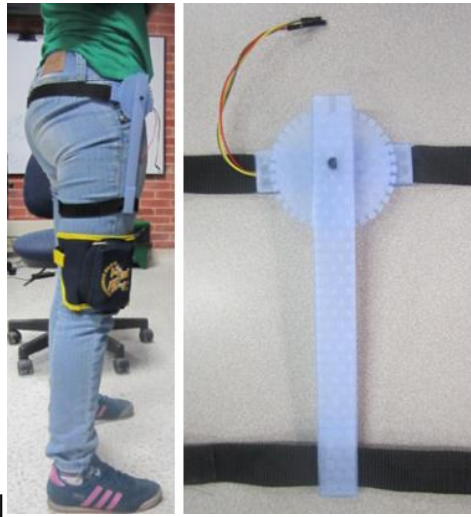
El elemento de medición se diseña en SolidWorks para posteriormente imprimirlas en 3D. Se obtienen tres piezas, una regleta móvil, la cual se adapta al segmento móvil, en este caso el muslo, una regla con los ángulos de referencia, en forma de transportador, con una resolución de 15°, y una última regleta que se fija a un punto de referencia constante, para mayor exactitud en la medición, en la Figura 24 se presentan las piezas ya mencionadas.



**Figura 24 SolidWorks, Piezas, Instrumento de prueba**

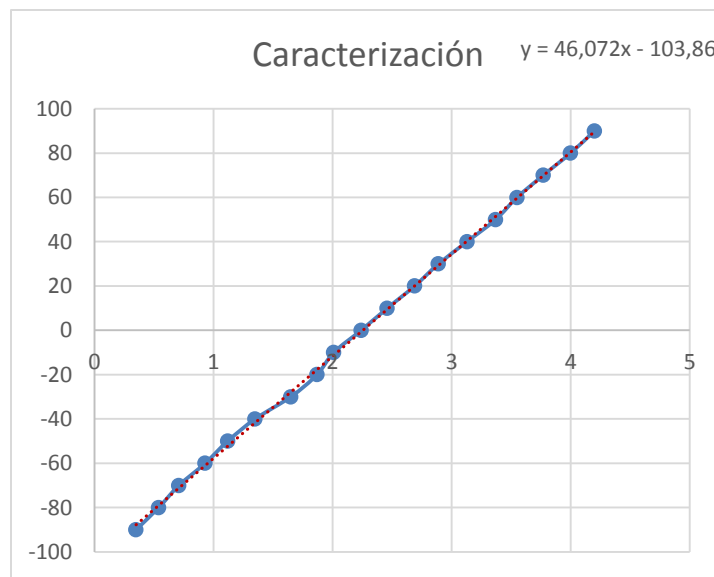
Para realizar la prueba se colocan las dos regletas en forma de eje de coordenadas y la pieza circular haciendo la función de transportador, como eje se ubica un potenciómetro que se usa para conocer la rotación de la pierna como se presenta en la Figura 25.

<sup>20</sup> [http://i01.i.aliimg.com/img/pb/176/579/386/386579176\\_827.jpg](http://i01.i.aliimg.com/img/pb/176/579/386/386579176_827.jpg)



**Figura 25 Ubicación y funcionamiento del instrumento de prueba.**

Se realiza la caracterización del sensor con un voltaje de encendido de 5,21v y se obtienen los siguientes datos, presentados en la Figura 28.



**Figura 28 Datos de caracterización de la herramienta de validación**

Se observa que la caracterización del sensor tiene una tendencia lineal, y la razón de cambio entre los datos es aproximadamente de 10°.

Se procede hacer una comparación entre los datos que arrojan la herramienta de validación y el sensor inercial, con respecto al movimiento de flexión/extensión del miembro inferior, implementado como se muestra en la Figura 29 a.

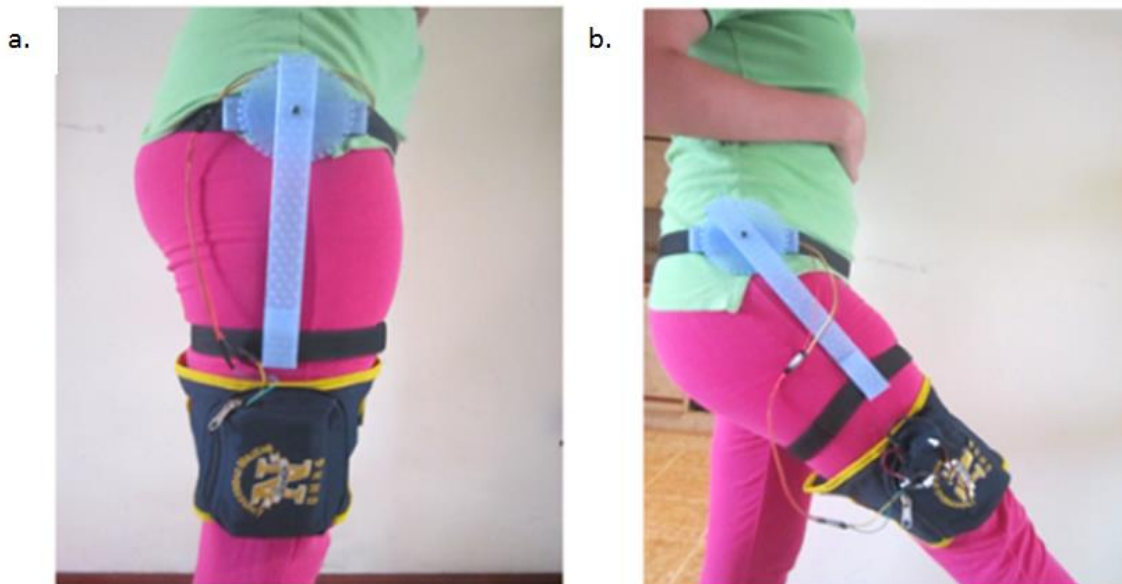


Figura 29 a. Implementación de la validación, b. Prueba de validación de datos

Se presenta una gráfica en la Figura 30, en la cual se observa los datos obtenidos de la herramienta de validación y el sensor inercial para un ángulo de aproximadamente 50° de rotación, ejecutando un movimiento de flexión, con el miembro inferior derecho, como se observa en la Figura 29 b.

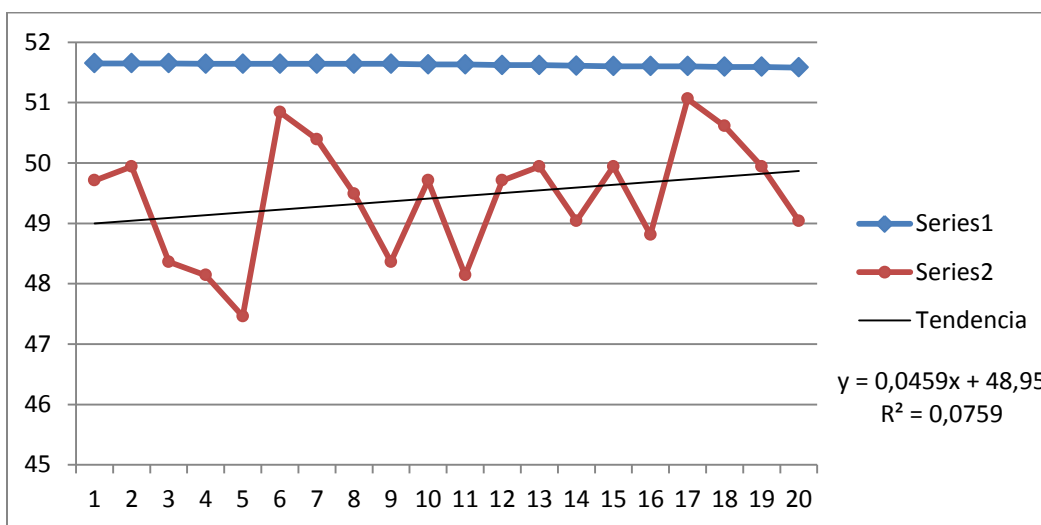


Figura 30 Validación de datos del acelerómetro contra el instrumento de medición tipo goniómetro

De esta grafica se puede observar, que el error entre los datos del sensor y los datos de la herramienta de validación, esta entre rangos aceptables ya que es mínimo para la amplitud de movimiento que se desea capturar.

Con el fin de observar los datos de una forma más detallada, se agrega una tabla de datos, que contiene las muestras tomadas de la herramienta de validación, los datos a partir de la línea de tendencia calculada, las muestras de los datos del sensor y para concluir el error absoluto entre los elementos a evaluar y el error promedio en porcentajes.

**Tabla 4 Comparación de datos**

Datos Herramienta	Datos Herramienta, Tendencia	Datos Sensor	% Error	Error Promedio (%)
49,71	48,9959	51,65	5,42	4,59
49,94	49,0418	51,65	5,32	
48,36	49,0877	51,65	5,22	
48,14	49,1336	51,64	5,10	
47,46	49,1795	51,64	5,00	
50,84	49,2254	51,64	4,91	
50,39	49,2713	51,64	4,81	
49,49	49,3172	51,64	4,71	
48,36	49,3631	51,64	4,61	
49,71	49,409	51,63	4,50	
48,14	49,4549	51,63	4,40	
49,71	49,5008	51,62	4,28	
49,94	49,5467	51,62	4,18	
49,04	49,5926	51,61	4,07	
49,94	49,6385	51,6	3,95	
48,81	49,6844	51,6	3,86	
51,06	49,7303	51,6	3,76	
50,61	49,7762	51,59	3,64	
49,94	49,8221	51,59	3,55	
49,04	49,868	51,58	3,43	

De la Tabla 4 se puede concluir que el error mínimo entre los datos es del 4,42 % y el máximo es del 5,42%, para un promedio del 4,59% de error.

Ahora se practica una validación a partir de comparación de los datos del sensor y la herramienta (Goniómetro) en un movimiento de una oscilación que representa la flexión del miembro inferior hasta 45° y retroceso a 0°. Se presenta una gráfica en la Figura 25, que representa los datos obtenidos.

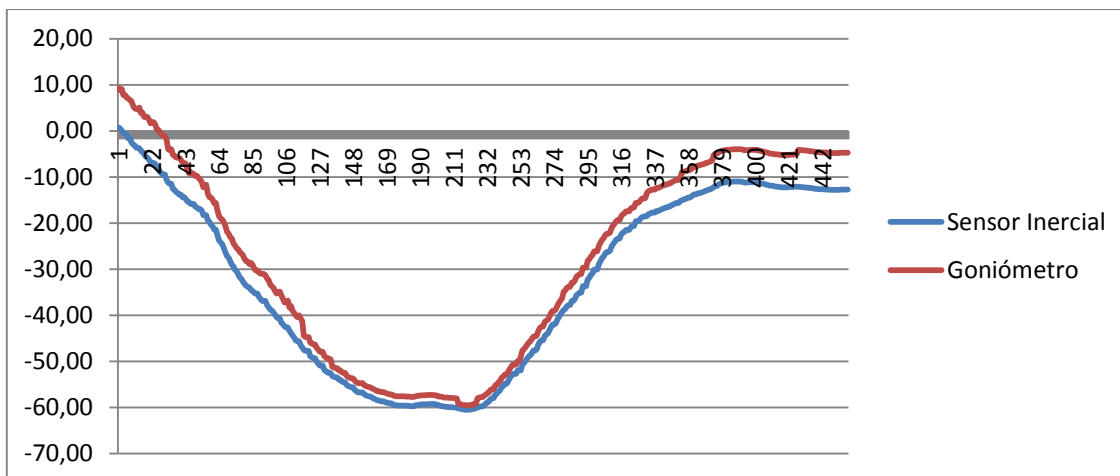


Figura 25 Validación de datos del acelerómetro contra el instrumento de medición tipo goniómetro en movimiento

Donde se puede validar que el sensor inercial es apto para monitorear el movimiento del miembro inferior, y desarrollar aplicaciones enfocadas a la ejercitación y rehabilitación del mismo.

#### 4.3.1 Encuestas

Para validar la percepción de los potenciales usuarios se realiza una encuesta que provea información con el objetivo de dar a conocer si el dispositivo está cumpliendo con sus objetivos. La encuesta se realiza a 11 personas diferentes; 5 son deportistas, 2 son especialistas en salud ocupacional, y 3 son personas que realizan actividades cotidianas.

- ¿Le gusta hacer ejercicio? Respuesta: Todos respondieron de forma afirmativa

El 100% de las personas encuestadas se sienten a gusto realizando ejercicio. Al ser una pequeña muestra no representa el total de la población. El fin de realizar la encuesta en su mayoría a deportistas (Karate-do, Taekwondo), es porque al ser un deporte de contacto pueden sufrir lesiones con mucha más facilidad.

- Indique su género. Respuesta 6 hombres y 5 mujeres
- ¿Cuál es la principal razón para hacer ejercicio? Respuesta:

Gusto	9	81.8%
Necesidad	1	9.1%
Relajación	1	9.1%
Apariencia	0	0%
Otro	0	0%

- Reconoce la importancia y cuidados que se debe tener con el miembro inferior (pierna). Respuesta: 8 personas si, 3 no.



Como la mayoría de encuestados son deportistas, un 72.7% del total de los encuestados reconocen la importancia y cuidados que se deben tener con el miembro inferior. Por otro lado se podría afirmar que 1 de cada 4 personas no deportistas, no saben la importancia ni los cuidados que deben tener con el miembro inferior.

- ¿Cuál es el lugar donde realiza los ejercicios? Respuesta

Gimnasio	4	36.4%
Centro médico	0	0%
Casa	3	27.3%
Trabajo	0	0%
Otro	4	36.4%

El 72.8% de la población encuestada realiza sus ejercicios en un lugar fuera de su casa (Gimnasio, otro), el 27.3% restante realiza actividad física en su casa. El 36.4 % restante manifestó que realiza actividad física en lugares como: parques, universidad y dojo (lugar de entrenamiento de Karate-do).

- ¿Después de utilizar el dispositivo considera divertida esta forma de hacer ejercicio? Respuesta: Todos respondieron de forma afirmativa
- Al utilizar el dispositivo indique en qué elemento se enfocó más en:

Hacer los ejercicios	4	36.4%
Hacer la mayor cantidad de puntos	3	27.3%
Ninguna	0	0%
Ver el movimiento del avatar	2	18.2%
Otro	3	27.3%

El 36.4 % de la población encuestados aseguro que el elemento en el que más se enfocó fue en hacer los ejercicios, el 27.3% sintió mayor atracción por realizar la mayor cantidad de puntos, el 18.2 % de las personas encuestadas sintió curiosidad y se enfocó más en el movimiento del avatar.

- ¿Tendría interés en utilizar un dispositivo móvil para realizar ejercicios con ambientes interactivos en el computador? Respuesta: Todos respondieron de forma afirmativa

Estas son las sugerencias que presentaron los encuestados:

- Aumentar la frecuencia del ejercicio.
- Incluir la articulación en el momento de la ejecución.
- Mejoras en el ambiente del juego.
- Más y mejores gráficos.
- Ninguna

De la encuesta se concluye que el dispositivo tiene un alto porcentaje de aceptación entre a población, de la cual gran porcentaje considera que es adecuado para hacer ejercicio y otro porcentaje considerable opina que es útil para la rehabilitación, ya que conoce la importancia de los cuidados del miembro inferior.

# CAPÍTULO 5: CONCLUSIONES

El dispositivo final tiene como objetivo medir únicamente la posición angular del movimiento que se realice con el miembro inferior, sin embargo con los sensores que tiene adaptados el dispositivo es posible obtener la aceleración y velocidad con la que se ejerce el movimiento, lo cual permite mayor grado de libertad al realizar el análisis del movimiento del jugador mediante la gráfica, ya que ésta, está formada por una curva parabólica la cual tiene como variables dependientes entre otros la velocidad y aceleración, y estos parámetros son previamente configurados en el juego, dejando así como única variable a medir, la posición.

Teniendo en cuenta la caracterización y anatomía del miembro inferior se seleccionó un ejercicio con el fin de integrar varios movimientos entre estos flexión/ extensión y aducción /abducción, movimientos que fueron determinantes para la selección del tipo de captura de movimiento y su correspondiente transmisión.

Los materiales, dispositivos electrónicos y código desarrollado para el diseño e implementación del prototipo final, fueron seleccionados utilizando tecnología de bajo costo y software libre, todo con el fin de resaltar la fácil adquisición y el alto impacto que podría generar en la recuperación de población con traumatología en el miembro inferior así mismo en su adquisición por parte de diferentes organizaciones como por ejemplo: prestadoras de salud, fisioterapeutas entre otras.

La interfaz del juego fue desarrollada bajo Processing, a pesar de ser un lenguaje de bajo nivel, este permitió desarrollar por completo un juego base de control y monitoreo de un movimiento para el miembro inferior, interfaz que interpreta y representa gráficamente en el ambiente de realidad virtual los datos capturados por el dispositivo, interactuando y realimentando al jugador/paciente.

Se comprobó que el dispositivo e interfaz funcionan y operan en su totalidad, por el contrario fue necesario modificar el método de amarre del dispositivo al muslo del jugador, el primer método presentaba problemas, por esa razón se tomó la decisión de cambiarlo.

Con respecto a la opinión de la gente encuestada, el dispositivo causa gran interés en cuanto se refiere a la forma interactiva de hacer ejercicio. Por otro lado las personas piensan que es totalmente factible y viable realizar procesos de rehabilitación con ayuda del dispositivo, además de que causa gran impresión en cuanto al manejo biomecánico que presenta; Por otro lado aconsejan mejorar los gráficos de la interfaz y agregar la articulación de la pierna

Con la validación de este primer prototipo, el cual demuestra que es factible hacer el monitoreo del movimiento del miembro inferior por medio de tecnología de bajo costo.

Debido a lo presentado previamente, los trabajos futuros estarán concentrados en aumentar los grados de libertad, inmersión e interacción en el exergame para proveer los datos que permitan mejorar la experiencia y el desempeño físico.

# Bibliografía

- [1] J. Sinclair, P. Hingston y M. Masek, «Consideraciones para el diseño de exergames,» *ACM Nueva York*, vol. ISBN, nº 978-1-59593-912-8 , pp. 289-295, 2007.
- [2] «SGA Serious Games Association,» Serious Games Association, 2015. [En línea]. Available: <http://translate.google.com.co/translate?hl=es-419&sl=en&u=http://www.seriousgamesassociation.com/games-for-health/&prev=search>. [Último acceso: 15 03 2015].
- [3] Burdea y Grigore, «Virtual rehabilitation-benefits and challenges.,» de *Virtual Reality Rehabilitation (Mental Health, Neurological, Physical, Vocatioal)*, Lausanne, Switzerland, 2002.
- [4] Jäger, M. A. Bundesanstalt für , S. y P. Tuula Solasaari , «Prevención de trastorno musculoesqueléticos en el lugar de trabajo.,» Jäger, PD Dr-Ing Matthias, Bundesanstalt für Arbeitsschutz, Dipl-Ing Ulf Steinberg, and Tuula Solasaari Pekki., Institut finnois de médecine du travail, 2004.
- [5] F. Foerster y J. Fahrenberg, «Motion pattern and posture: Correctly assessed by calibrated accelerometers,» *Springer*, vol. 32, pp. 450-457, 2000.
- [6] E. Jovanov, A. Milenkovic, C. Otto y P. C. De Groen, «A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation,» *Journal of NeuroEngineering and rehabilitation*, vol. 2, nº 1, p. 6, 2005.
- [7] R. E. Mayagoitia, A. V. Nene y P. H. Veltink, «Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems,» *Journal of biomechanics*, vol. 35, nº 4, pp. 537-542, 2002.
- [8] Microsoft, «Xbox Kinect,» Microsoft, [En línea]. Available: <http://www.xbox.com/en-US/xbox-360/accessories/kinect>. [Último acceso: 26 05 2015].
- [9] C. Wingrave, B. Williamson, P. Varcholik, J. Rose, A. Miller, E. Charbonneau y J. J. LaViola, «The Wiimote and Beyond: Spatially Convenient Devices for 3D User Interfaces,» *IEEE Computer Graphics and Applications*, vol. 30, pp. 71-85, 2020.
- [10] E. Suma, B. Lange , A. Rizzo, D. Krum y M. Bolas, « FFAST:The Flexible Action and Articulated

- Skeleton Toolkit,» *IEEE*, pp. 247-248, 2011.
- [11] P. Rakprayoon, M. Ruchanuruks y A. Coundoul, «Kinect-based obstacle detection for manipulator.,» *2011 IEEE/SICE International Symposium on IEEE*, pp. 68-73, 2011.
- [12] S. Matyunin, D. Vatolin, Y. Berdnikov y M. Smirnov, «Temporal filtering for depth maps generated by Kinect depth camera.,» *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1-4, 2011.
- [13] V. Frati y D. Prattichizzo, «Using Kinect for hand tracking and rendering in wearable haptics,» *World Haptics Conference (WHC), 2011 IEEE*, pp. 317-321, 2011.
- [14] J. Raheja, A. Chaudhary y K. Singal, «Tracking of Fingertips and Centers of Palm Using KINECT.,» *Computational Intelligence, Modelling and Simulation (CIMSIM)*, pp. 248-252, 2011.
- [15] J. Stowers, M. Hayes y A. Bainbridge-Smith, «Altitude control of a quadrotor helicopter using depth map from Microsoft Kinect sensor,» *Mechatronics (ICM), 2011 IEEE* , pp. 358-362, 2011.
- [16] D. Bradshaw y K. Ng, «Tracking Conductors Hand Movements Using Multiple Wiimotes,» *IEEE Computer Society.*, pp. 93-99, 2008.
- [17] P. Varcholik, D. Barber y D. Nicholson, «INTERACTIONS AND TRAINING WITH UNMANNED SYSTEMS AND THE NINTENDO WIIMOTE,» *Institute of Simulation \& Training University of Central Florida.*, 2008.
- [18] J. Decker, H. Li, D. Losowyj y V. Prakash, «Wiihabilitation: Rehabilitation of Wrist Flexion and Extension Using a Wii Remote controller-Based Game System.,» *www.docstoc.com*, 2009.
- [19] T. Leyvand, C. Meekhof, Y. Wei, J. Sun y B. Guo, «Kinect Identity: Technology and Experience,» *Computer*, vol. 44(4), pp. 94-96, 2011.
- [20] A. Da Gama, T. Chaves , L. Figueiredo y V. Teichrieb, «Poster: Improving motor rehabilitation process through a natural interaction based system using Kinect sensor,» *3D User Interfaces (3DUI), 2012 IEEE* , pp. 145-146, 2012.
- [21] J. Ekelmann y B. Butka, « Kinect controlled electro-mechanical skeleton. Southeastcon,» *Proceedings of IEEE*, pp. 1-5, 2012.
- [22] M. N. Center, «Kinect Effect Reaches Into Hospitals, Senior Centers,» Microsoft, 19 Dec 2001. [En línea]. Available: <http://www.microsoft.com/en-us/news/features/2011/dec11/12-19KinectEffect.aspx>. [Último acceso: 02 02 2015].

- [23] Reflexion Health, «Reflexion Health's Rehabilitation Measurement Tool is Advancing Physical Therapy with Kinect for Windows,» *Reflexion Health*, 2013.
- [24] K. & H. D. Werbach, «For the Win: How Game Thinking Can Revolutionize Your Business,» Wharton Digital Press, 2012.
- [25] J. Burke, M. McNeill, D. Charles, P. Morrow, J. rosbie, S. McDonough y S. McDonough, «Augmented reality games for upper-limb stroke rehabilitation.,» *Games and Virtual Worlds for Serious Applications(VS-GAMES)*, vol. Second international Conference, pp. 75-78, 2012.
- [26] V. I. Pavlovic, R. Sharma y T. S. Huang, «Visual interpretation of hand gestures for human-computer interaction: A review. Pattern Analysis and Machine Intelligence,» *IEEE Transactions*, vol. 19, nº (7), pp. 677-695, (1997).
- [27] W. & W. Ourique De Morais, «N A Serious Computer Game to Assist Tai Chi Training for the Elderly,» de *2011IEEE 1st International Conference on Serious Games and Applications for Health, SeGAH 2011*, 2011.
- [28] O. Assad, R. Hermann, D. Lilla, B. Mellies, R. Meyer, L. Shevach y others, «Motion-based games for parkinson's disease patients. En Entertainment Computing--ICEC 2011,» *Springer*, pp. 47-58, 2011.
- [29] S. Jitaree, A. Phinyomark, H. Hu, P. Phukpattaranont y C. Limsakul, «DESIGN OF EMG BIOFEEDBACK SYSTEM FOR LOWER-LIMB EXERCISES OF THE ELDERLY USING VIDEO GAMES».
- [30] A. Michalski, C. Glazebrook, A. Martin, W. Wong, A. Kim, K. Moody y others, «Assessment of the postural control strategies used to play two Wii Fit™ videogames,» *Gait & Posture*, vol. 36, nº 3, pp. 449-453, 2012.
- [31] D. B. Chaffin, A. Gunnar y J. M. Bernard, «Occupational biomechanics,» *New York: Wiley*, 1999.
- [32] J. Medrano, «DISEÑO DE UN EXOESQUELETO DE MIEMBROS INFERIORES PARA LA MOVILIDAD DE PERSONAS CON PARAPLEJÍA,» de *Copyright © 2014 by AMRob Memorias del XVI Congreso Mexicano de Robótica 2014*, Mazatlán, Sinaloa, México, 2014.
- [33] C. H. Guzmán Valdivia, . A. Blanco Ortega, M. A. Oliver Salazar y H. R. Azcaray Rivera, «Modelado y Simulación de un Robot Terapéutico,» *Revista Ingeniería Biomédica*, vol. 7, nº ISSN 1909-9762 , pp. 42-50, 2013.
- [34] B. William, *Mediciones y pruebas Eléctricas y Electrónicas*, Barcelona : Marcombo S.A., 1995.

- [35] R. E. Kalman, «A New Approach to Linear Filtering and Prediction Problems,» *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, pp. 35-45 , 1960.
- [36] I. B. Montero, Operaciones auxiliares de montaje de componentes informáticos, Madrid, España: Paraninfo, 2010 .
- [37] D. M. Galvez Dominguez, M. A. Arcas Patricio, J. C. Leon Castro y M. A. Elosegui Bilbao, Fisioterapeuta del servicio de salud de la comunidad de madrid., España: MAD, 2005.
- [38] A. Barrientos, Fundamentos de Robótica, McGraw Hill, 2007.
- [39] B. Gorla y M. Renaud, Modèles des Robots Manipulateurs: Application à leur Commande, Automatisation \& Production, 1984.
- [40] J. Denavit y R. Hartenberg, Kinematic Synthesis of Linkages, McGraw Hill, 1964.
- [41] D. M. Galvez Dominguez, M. A. Arcas Patricio, J. C. . Leon Castro y . J. L. Elosegui Bilbao, Fisioterapeuta del servicio de salud de la comunidad de madrid., ESPAÑA: MAD, 2005.



# Apéndice 1 Conexión Arduino – Sensor Inercial

```
/* =====
// https://github.com/jrowberg/i2cdevlib
=====*/

#include "Wire.h"

#include "I2Cdev.h"

#include "MPU6050_6Axis_MotionApps20.h"

MPU6050 sensor;

// control sensor /variables de estado
bool dmpListo = false; // true => DPM inicio exitoso
uint8_t sensorIntEstado; // Estado del byte de interrupción del sensor
uint8_t devEstado; // estado del dispositivo despues de cada
operación
// (0 = exitoso, !0 = error)
uint16_t TamPaq; // tamaño esperado del paquete (por defecto 42 bytes)
uint16_t fifoCon; // conteo de bytes actuales en el FIFO
uint8_t fifoBuffer[64]; // Buffer de almacenamiento FIFO

// orientación/variables de posicion
Quaternion q; // [w, x, y, z] contenedor del cuaternio
VectorFloat gravedad; // [x, y, z] vector de gravedad
float angulos[3]; // [yaw, pitch, roll] yaw/pitch/roll vector
de gravedad
float ax, ay, az, gx, gy, gz, mx, my, mz; //variables en los tres ejes
//aceleracion, giroscopio,
magnetometro
// =====
// === INTERRUPT DETECTION ROUTINE ===
// =====

volatile bool InterSensor = false; // pin de interrupción en estado alto
void dmpDatosListos() {
    InterSensor = true;
}

// =====
// === INITIAL SETUP ===
// =====

void setup() {

    Wire.begin();

    // incializar comunicación serial
```

```

Serial.begin(115200);

// inicializar dispositivo
Serial.println(F("Inicializar Sensor"));
sensor.initialize();

// verificar conexión
Serial.println(F("Provando conexión"));
Serial.println(sensor.testConnection() ? F("conexión Exitosa") :
F("Falla de conexión"));

// esperar para comenzar
Serial.println(F("\nSend any character to begin DMP programming and
demo: "));
while (Serial.available() && Serial.read()); // Buffer vacío
while (!Serial.available()); // esperar por datos
while (Serial.available() && Serial.read()); // Buffer vacío de nuevo

// cargar y configurar el DMP
Serial.println(F("Inicializar DMP..."));
devEstado = sensor.dmpInitialize();

// Asegurar de que está trabajando (devuelve 0 si no)
if (devEstado == 0) {
    // Encender el DMP, ya está listo
    Serial.println(F("Habilitar DMP..."));
    sensor.setDMPEnabled(true);

    // Habilitar la detección de la interrupción del Arduino
    Serial.println(F("Interrupción INT0 Habilitada"));
    attachInterrupt(0, dmpDatosListos, RISING);
    sensorIntEstado = sensor.getIntStatus();

    // Colocar en alto la bandera que indica DMP listo
    Serial.println(F("DMP Listo! Esperar por interrupción"));
    dmpListo = true;

    // Obtener el tamaño del paquete DMP
    TamPaq = sensor.dmpGetFIFOPacketSize();
} else {
    // ERROR!
    // 1 = Fallo en la carga inicial de la memoria
    // 2 = Fallo en actualización de configuración del DMP

    Serial.print(F("Falla DMP(code "));
    Serial.print(devEstado);
    Serial.println(F(")"));
}

}

// =====

```

```

// ===                                MAIN PROGRAM LOOP                                ===
// =====

void loop() {
  // Si dmpListo = 0, el programa para
  if (!dmpListo) return;

  // Esperar por la interrupción del sensor
  // o un paquete extra disponible
  while (!InterSensor && fifoCon < TamPaq) {
  }

  // Inicializar el valor de la Bandera de interrupción
  InterSensor = false;
  sensorIntEstado = sensor.getIntStatus();

  // Obtener numero de bytes actuales en el FIFO
  fifoCon = sensor.getFIFOCount();

  // Comprobar si hay desbordamiento
  if ((sensorIntEstado & 0x10) || fifoCon == 1024) {
    // Reiniciar
    sensor.resetFIFO();
    Serial.println(F("desbordamiento FIFO!"));
  }

  // Si no, verificar los datos del DMP
  } else if (sensorIntEstado & 0x02) {
    // esperar por la longitud correcta del paquete del FIFO
    while (fifoCon < TamPaq) fifoCon = sensor.getFIFOCount();

    // Leer el paquete del FIFO
    sensor.getFIFOBytes(fifoBuffer, TamPaq);

    fifoCon -= TamPaq;

    MadgwickQuaternionUpdate(ax, ay, az, gx, gy, gz, mx, my, mz)
    sensor.dmpGetQuaternion(&q, fifoBuffer);
    sensor.dmpGetGravity(&gravedad, &q);
    sensor.dmpGetYawPitchRoll(angulos, &q, &gravedad);
    Serial.print("flexion - Extension");
    Serial.print("\t");
    Serial.print(angulos[2] * 180/M_PI);
    Serial.print("\t");
    Serial.print("Aduccion - Abduccion");
    Serial.print("\t");
    Serial.println(angulos[1] * 180/M_PI);
  }
}

```

## Apéndice 2 Configuración comandos AT

```
//Librería para establecer comunicación serie en
```

```
//pines no predeterminados
#include <SoftwareSerial.h>

//pines RXD,TDX para comunicacion con el módulo Bluetooth.
SoftwareSerial serial2(6,7); //6 RX, 7 TX.

void setup()
{
  //Establece velocidad de transmision con el PC
  Serial.begin(9600);
  //Establece velocidad de transmision con
  //el modulo Bluetooth
  serial2.begin(38400);
  Serial.println("Cofiguracion por comandos AT: ");
}

void loop()
{
  if(Serial.available())
  {
    serial2.write(Serial.read());
  }
  if(serial2.available())
  {
    Serial.write(serial2.read());
  }
}
```

# Apéndice 3 Conexión Arduino - sensor Inercial - Bluetooth

```
/* =====
// I2C device class (I2Cdev) demonstration Arduino sketch for MPU6050
// class using DMP (MotionApps v2.0)
// 6/21/2012 by Jeff Rowberg <jeff@rowberg.net>
// Updates should (hopefully) always be available at
// https://github.com/jrowberg/i2cdevlib
=====*/

/* =====
I2Cdev device library code is placed under the MIT license
Copyright (c) 2012 Jeff Rowberg
Permission is hereby granted, free of charge, to any person obtaining a
copy
of this software and associated documentation files (the "Software"), to
deal
in the Software without restriction, including without limitation the
rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included
in
all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
=====*/

#include "Wire.h"

#include "I2Cdev.h"

#include "MPU6050_6Axis_MotionApps20.h"

#include <SoftwareSerial.h> //Librería que permite establecer
                          //el puerto de omunicación serie en otros
pins

//Conectamos los pins RXD,TDX del módulo Bluetooth.
SoftwareSerial BT(6,7); //6 RX, 7 TX.
```

```

MPU6050 mpu;

#define OUTPUT_READABLE_YAWPITCHROLL

#define LED_PIN 13 //Pin indicador de actividad
bool blinkState = false;

// control MPU /variables de estado
bool dmpReady = false; // true => DPM inicio exitoso
uint8_t mpuIntStatus; // Estado del byte de interrupción del MPU
uint8_t devStatus; // estado del dispositivo despues de cada
operación

// (0 = exitoso, !0 = error)
uint16_t packetSize; // tamaño esperado del paquete (por defecto 42
bytes)
uint16_t fifoCount; // conteo de bytes actuales en el FIFO
uint8_t fifoBuffer[64]; // Buffer de almacenamiento FIFO

// orientación/variables de posición
Quaternion q; // [w, x, y, z] contenedor del cuaternio
VectorFloat gravity; // [x, y, z] vector de gravedad
float ypr[3]; // [yaw, pitch, roll] yaw/pitch/roll vector de
gravedad

// =====
// === INTERRUPT DETECTION ROUTINE ===
// =====

volatile bool mpuInterrupt = false; // pin de interrupción en estado
alto
void dmpDataReady() {
    mpuInterrupt = true;
}

// =====
// === INITIAL SETUP ===
// =====

void setup() {

    Wire.begin();

    // incializar comunicación serial
    BT.begin(115200);

    // inicializar dispositivo
    BT.println(F("Initializing I2C devices..."));
    mpu.initialize();

    // verificar conexión
    BT.println(F("Testing device connections..."));
    BT.println(mpu.testConnection() ? F("MPU6050 connection
successful") : F("MPU6050 connection failed"));
}

```

```

    // esperar para comenzar
    BT.println(F("\nSend any character to begin DMP programming and demo:
"));
    while (BT.available() && BT.read()); // Buffer vacio
    while (!BT.available());           // esperar por datos
    while (BT.available() && BT.read()); // Buffer vacio de nuevo

    // lCrgar y configurar el DMP
    BT.println(F("Initializing DMP..."));
    devStatus = mpu.dmpInitialize();

    // Asegurar de que esta trabando (devuelve 0 si no)
    if (devStatus == 0) {
        // Encender el DMP, ya esta listo
        BT.println(F("Enabling DMP..."));
        mpu.setDMPEnabled(true);

        // Habilitar la detección de la interrupcion del Arduino
        BT.println(F("Enabling interrupt detection (Arduino external
interrupt 0)..."));
        attachInterrupt(0, dmpDataReady, RISING);
        mpuIntStatus = mpu.getIntStatus();

        // Colocar en alto la bandera que indica DMP listo
        BT.println(F("DMP ready! Waiting for first interrupt..."));
        dmpReady = true;

        // Obtener el tamaño del paquete DMP
        packetSize = mpu.dmpGetFIFOPacketSize();
    } else {
        // ERROR!
        // 1 = Fallo en la carga inicial de la memoria
        // 2 = Fallo en actualización de configuracion del DMP

        BT.print(F("DMP Initialization failed (code ");
        BT.print(devStatus);
        BT.println(F(")"));
    }

    // Configurar el LED como salida
    pinMode(LED_PIN, OUTPUT);
}

// =====
// ===                               MAIN PROGRAM LOOP                               ===
// =====

void loop() {
    // Si dmpReady = 0, el programa para
    if (!dmpReady) return;

    // Esperar por la interrupción del MPU

```

```

    // o un paquete extra disponible
    while (!mpuInterrupt && fifoCount < packetSize) {
    }

    // Inicializar el valor de la Bandera de interrupción
    mpuInterrupt = false;
    mpuIntStatus = mpu.getIntStatus();

    // Obtener numero de bytes actuales en el FIFO
    fifoCount = mpu.getFIFOCount();

    // Comprobar si hay desbordamiento
    if ((mpuIntStatus & 0x10) || fifoCount == 1024) {
        // Reiniciar
        mpu.resetFIFO();
        BT.println(F("FIFO overflow!"));

    // Si no, verificar los datos del DMP
    } else if (mpuIntStatus & 0x02) {
        // esperar por la longitud correcta del paquete del FIFO
        while (fifoCount < packetSize) fifoCount = mpu.getFIFOCount();

        // Leer el paquete del FIFO
        mpu.getFIFOBytes(fifoBuffer, packetSize);

        fifoCount -= packetSize;

#ifdef OUTPUT_READABLE_YAWPITCHROLL
        // Mostrar angulos Euler en grados
        mpu.dmpGetQuaternion(&q, fifoBuffer);
        mpu.dmpGetGravity(&gravity, &q);
        mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
        BT.println(ypr[2] * 180/M_PI);
        BT.println(ypr[1] * 180/M_PI);
#endif

        // Encender el led
        blinkState = !blinkState;
        digitalWrite(LED_PIN, blinkState);
    }
}

```



# Apéndice 4 Processing: Recepción de datos

```
import processing.serial.*;

//Puerto Serial
Serial port; //Objeto para acceder al puerto
//Variables de transmision
String dato; //Recibe los datos del puerto
String palabra = "Send any character to begin DMP programming and demo:";
int total=0,cont=0; //contador de Bytes
float grados=0; //Variable de movimiento

void setup()
{
  print(Serial.list()); //Imprime la lista de puertos disponibles
  port = new Serial(this, "COM4", 115200); //Configuro el puerto y la
  velocidad
}

void draw()
{
}

//Interrupcion del Puerto Serial
void serialEvent(Serial port)
{
  //Leer dato del puerto
  dato = port.readStringUntil('\n');
  //Verifica que no este vacio
  if (dato != null)
  {
    total = dato.length(); //Obtener tamaño del dato
  }
  //Obtener palabra
  if (total==56)
  {
    for(int c=0; c<56; c++)
    {
      if (c<53)
      {
        if (dato.charAt(c)==palabra.charAt(c)) cont++;
      }
    }
    total = 2;
    //Comenzar la transmisión de datos
    if (cont==53)
    {
      port.write('r');
    }
  }
  //Obtener los datos del ángulo Pitch
  if (total >= 6 && total <=8)
  {
    grados = Float.parseFloat(dato)-8;
```

```
println("pitch");
println(grados);
total = 2;
}
//Obtener los datos del ángulo Yaw
if (total >= 9 && total <=11)
{
    grados = Float.parseFloat(dato)-8;
    grados = grados/1000;
    println("yaw");
    println(grados);
    total = 2;
}
}
```

## Apéndice 5 Processing: Interfaz y juego

```
//****Librerias necesarias****//
import processing.serial.*;
import processing.opengl.*;

//*****Variables para la transmision**//
//Puerto Serial
Serial port;
//transmision
String dato;
String palabra = "Send any character to begin DMP programming and demo:";
int total=0,cont=0;
//Ventanas
PGraphics VScore,VScore1;

//*****Interfaz Estatica*****//
//Tamaño de la aplicacion
int alto = 600, ancho=1200;
//Color de fondo de la aplicacion
color negro = 0;
//Posicion de los ceros en el sistema cordenado
int x=40, y=255;

//+++++++Interfaz Dinamica*****//
////////Variables////////
//Pierna
float grados=0; //recibe el dato del sensor
//Pelota
float a=9.8, pct=0, step=0.01;
float X=0,Y=0,Xo=90,Yo=200;
float Vox=326, Voy=20;
int c=0;
//Señal
float dx=1.0;
float[] valor_x;
float[] valor_y;
int contador=1;
//Reporte
float[] datos;
int[] N_Grafica;
int contDatos=0;
int contGrafica=0;
float[][]graficas;
float aux;
//Puntaje
int puntos=0;
//Boton
color boton;
```

```

int N_boton=0; //identificar el boton sobre el cual esta el mouse
//////////Estados//////////
int InitJuego; /*Iniciar juego*/
int RecivDato; //Decide si guardar el dato que llega
int PieToBal; //Detecta cuando el pie toca el balón
int MosScore; //el boton score esta presionado
int canasta; //el balón toca la canasta
int MosReport; //El boton reporte esta presionado

```

```

void setup() {

    size(ancho, alto, OPENGL);
    VScore = createGraphics(100, 40);
    VScore1 = createGraphics(300, 300);
    lights();
    smooth();
    String PortName = Serial.list()[0];
    println(PortName);
    //port = new Serial(this, "COM6", 115200);
    valor_x = new float[240];
    valor_y = new float[240];
    datos = new float[240];
    N_Grafica = new int[4];
    graficas = new float[240][4];
    InitJuego = 0;
    RecivDato = 0;
    PieToBal = 0;
    MosScore = 0;
    MosReport = 0;
}

```

```

void draw()
{
    background(negro);
    noStroke();
    InterStatic();
    if (InitJuego==1)
    {
        Jugar();
    }
    else
    {
        Juego();
    }
    if(MosScore==1 && InitJuego==0)
    {
        Score();
    }
    if(MosReport==1 && InitJuego==0)
    {
        Report();
    }
}

```

```

    }

}

//*****Funciones Intefaz estatica*****//

///Principal
void InterStatic ()
{
    Botones();
    GraficaMov();
}

///Botones
void Botones()
{
    textSize(15);
    fill(255, 0, 0, 240);
    text("PLAY", 42, 32);
    textSize(15);
    text("SCORE", 117, 32);
    text("REPORT", 197, 32);
    text("INSTRUCTIONS", 287, 32);

    stroke(255, 0, 0, 200);
    fill(0,0,255,60);
    rect(30, 10, 60, 30);
    rect(110, 10, 60, 30);
    rect(190, 10, 70, 30);
    rect(280, 10, 120, 30);

    AnimBoton();
}

///Grafica del movimiento
void GraficaMov()
{
    stroke(0, 0, 255, 200);
    fill(0,0,255,60);
    rect(420, 10, 240, 100);
}

///Graficos del juego
void Juego()
{
    noStroke();
    translate(x,y);
    fill(255, 0, 0, 200);
    ellipse(50,50,50,50);
    fill(0, 0, 255, 200);
    rect(40,75,20,70);
    fill(0, 255, 0, 100);
    rect(40,145,15,70);
}

```

```

fill(0, 0, 255, 200);
triangle(50, 100, 10, 165, 100, 165);
fill(255, 0, 0, 200);
ellipse(650,50,50,50);
fill(0, 0, 255, 200);
rect(640,75,20,70);
fill(0, 255, 0, 255);
rect(635,145,10,70);
fill(0, 255, 0, 255);
rect(655,145,10,70);
fill(255, 0, 0, 220);
rect(635,210,10,5);
fill(255, 0, 0, 220);
rect(655,210,10,5);
fill(0, 0, 255, 200);
triangle(650, 100, 610, 165, 700, 165);

if (InitJuego == 0 )
{
  pushMatrix();
  translate(45, 145, 0);
  fill(0, 255, 0, 200);
  rect(0, 0, 15, 70);
  translate(8, 67.5, 0);
  fill(255, 0, 0, 200);
  ellipse(0,0,16,5);
  popMatrix();
}

fill(255, 100, 0, 200);
quad(400, 215, 435, 215, 430, 230, 405, 230);
fill(255, 100, 0, 200);
quad(500, 215, 535, 215, 530, 230, 505, 230);

}

void Score()
{
  VScore.beginDraw();
  VScore.background(51);
  VScore.textSize(15);
  VScore.fill(0, 0, 255, 255);
  VScore.text("Score="+puntos, 0, 25);
  VScore.endDraw();
  image(VScore, 100, -10);
}

void Report()
{
  VScore1.beginDraw();
  VScore1.background(51);
  VScore1.stroke(255, 0, 0, 200);
  VScore1.fill(0,0,255,60);
  VScore1.rect(30, 10, 240, 280);

```

```

for(int i=0; i<datos.length; i++)
{
    VScore1.ellipse(i+30, graficas[i][0]+30, 2,2);
    VScore1.ellipse(i+30, graficas[i][1]+95,2,2);
    VScore1.ellipse(i+30, graficas[i][2]+160, 2,2);
    VScore1.ellipse(i+30, graficas[i][3]+225, 2,2);
}
VScore1.endDraw();
image(VScore1, 100, -10);
}

void correr()
{
    for(int i=0; i<N_Grafica[contGrafica]+1; i++)
    {
        aux=datos[0];
        for(int j=0; j<datos.length-1; j++)
        {
            datos[j] = datos[j+1];
        }
        datos[239]=aux;
    }
}

void guardar()
{
    for (int i=0; i<datos.length; i++)
    {
        graficas[i][contGrafica]=datos[i];
    }
}

//*****Funciones Intefaz Dinamica*****//
///Jugar
void Jugar()
{
    Juego();
    Pierna();
    Senal();
    if(grados == -15)
    {
        RecivDato = 0;
        PieToBal = 1;
    }
    Balon(PieToBal);
}

void Senal()
{
    if (contador == 240)contador=1;
    valor_x[contador] = valor_x[contador-1]+dx;
    valor_y[contador] = grados;
    for (int i=1; i<=valor_x[contador]; i++)
    {

```

```

        ellipse(valor_x[i]+380, valor_y[i]-195, 2,2);
    }
    if(PieToBal==0)
    {
        if(contDatos==240) contDatos=0;
        datos[contDatos]=valor_y[contador];
        contDatos++;
    }
    else
    {
        N_Grafica[contGrafica]=contDatos-1;
    }
    contador += 1;
}
///Movimiento de la pierna
void Pierna()
{
    if (grados < -15) grados=-15;
    if (grados > 45) grados = 45;

    pushMatrix();
    translate(45, 145, 0);
    rotateZ(radians(grados));
    fill(0, 255, 0, 200);
    rect(0, 0, 15, 70);
    translate(8, 67.5, 0);
    fill(255, 0, 0, 200);
    ellipse(0,0,16,5);
    popMatrix();
}

///dinamica del balon
void Balon(int estado)
{
    if (estado == 1)
    {
        pct += step;
        //println(pct);
        if (pct < 1.0)
        {
            X = Xo + Vox*pct;
            //println(X);
            Y = Yo + Voy*pct - 0.5*a*pct;
            //println(Y);
        }

        fill(255, 255, 0, 100);
        ellipse(X, Y, 25, 25);
        if (X>=405 && X<=425 && Y>=215)
        {
            c++;
            textSize(15);

```



```

fill(255, 0, 255, 240);
text("Bien!!!", 330, 115);
if (c>=60)
{
    InitJuego=0;
    println("termino");
    puntos += 100;
    c=0;
    println(datos);
    correr();
    println("ahora");
    println(datos);
    guardar();
    println(contDatos);
    println(contGrafica);
    println(N_Grafica);
    println(graficas[239][contGrafica]);
    contGrafica++;
    if (contGrafica == 4) contGrafica=0;

}
}
else
{
    fill(255, 255, 0, 100);
    ellipse(Xo, Yo, 25, 25);
}
}

```

```

///Animacion Botenes
void AnimBoton()
{
    if (OverBoton())
    {
        boton = color(0,0,255,100);
        switch(N_boton)
        {
            case 1:
            {
                fill(boton);
                rect(30, 10, 60, 30);
                break;
            }
            case 2:
            {
                fill(boton);
                rect(110, 10, 60, 30);
                break;
            }
            case 3:
            {
                fill(boton);

```

```

        rect(190, 10, 70, 30);
        break;
    }
    case 4:
    {
        fill(boton);
        rect(280, 10, 120, 30);
        break;
    }
}
}
}
}

```

```

//*****Interrupciones*****//

```

```

///Interrupcion cuando el mause esta sobre el boton
boolean OverBoton()

```

```

{
    //println(mouseX);
    //println(mouseY);
    if (mouseX > 30 && mouseX < 90 && mouseY > 10 && mouseY < 40)
    {
        N_boton = 1;
        if (mousePressed)
        {
            InitJuego = 1;
            RecivDato = 1;
            PieToBal = 0;
            grados=0;
            X=Xo;
            Y=Yo;
            pct=0;
        }
        //println(N_boton);
        return true;
    }
    else {
        if (mouseX > 110 && mouseX < 170 && mouseY > 10 && mouseY < 40)
        {
            N_boton = 2;
            if (mousePressed)
            {
                MosScore=1;
            }
            //println(N_boton);
            return true;
        }
        else{
            if (mouseX > 190 && mouseX < 260 && mouseY > 10 && mouseY < 40)
            {
                N_boton = 3;
                if (mousePressed)
                {

```

```

        MosReport=1;
    }
    //println(N_boton);
    return true;
}
else
{
    if (mouseX > 280 && mouseX < 400 && mouseY > 10 && mouseY < 40)
    {
        N_boton=4;
        return true;
    }
    else
    {
        if (mousePressed)
        {
            MosScore=0;
            MosReport=0;
        }
        N_boton=0;
        //println(N_boton);
        return false;
    }
}
}
}
}

```

///Interupcion cuando hay un dato en la transmision serial

```

void serialEvent(Serial port)
{
    dato = port.readStringUntil('\n');
    if (dato != null)
    {
        total = dato.length();
        //println(total);
        //println(dato);
    }
    if (total==56)
    {
        for(int c=0; c<56; c++)
        {
            if (c<53)
            {
                if (dato.charAt(c)==palabra.charAt(c)) cont++;
                //println(val.charAt(c));
            }
        }
        total = 2;
        println(cont);
        if (cont==53)
        {
            port.write('r');
        }
    }
}

```

```
}  
if (total >= 6 && total <=8)  
{  
    grados = Float.parseFloat(dato)-8;  
    //println(dato);  
    println(grados);  
    total = 2;  
}  
}
```

---