

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA BIOMÉDICA  
TRABAJO FIN DE GRADO**

**DISEÑO E IMPLEMENTACIÓN DE UN  
ENTORNO DE REALIDAD VIRTUAL PARA LA  
REHABILITACIÓN  
DE MARCHA ASISTIDA POR EL ROBOT  
CPWALKER**

**ENMIN ZHONG**

**2021**



## GRADO EN INGENIERÍA BIOMÉDICA

### TRABAJO FIN DE GRADO

**Título:** Diseño e implementación de un entorno de realidad virtual para la rehabilitación de marcha asistida por el robot CPWalker.

**Autor:** Enmin Zhong

**Tutor:** Dr. D. Eduardo Rocon de Lima

**Ponente:** Dr. D. Álvaro Gutiérrez Martín

**Departamento:** Tecnología Fotónica y Bioingeniería

### MIEMBROS DEL TRIBUNAL

**Presidente:** D. ....

**Vocal:** D. ....

**Secretario:** D. ....

**Suplente:** D. ....

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de:  
.....

Madrid, a                      de                      de 20...

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA BIOMÉDICA  
TRABAJO FIN DE GRADO**

**DISEÑO E IMPLEMENTACIÓN DE UN ENTORNO DE  
REALIDAD VIRTUAL PARA LA REHABILITACIÓN  
DE MARCHA ASISTIDA POR EL ROBOT CPWALKER**

**Autor**

**Enmin Zhong**

**Tutores**

**Dr. D. Álvaro Gutiérrez Martín**

**Dr. D. Eduardo Rocon de Lima**

**2021**



## RESUMEN

El objetivo de este Trabajo de Fin de Grado es diseñar e implementar un entorno de Realidad Virtual (RV) como marco para estudiar su papel en el campo de la neurorrehabilitación, especialmente en los niños con parálisis cerebral.

Para ello, se ha integrado un entorno virtual a la plataforma robótica CPWalker diseñado por el CSIC (Centro Superior de Investigaciones Científicas). Es el primer andador con apoyo dinámico del peso corporal y marcha activa diseñado para aplicarlo en entornos reales. La incorporación de la realidad virtual a la terapia asistida por robótica tiene un papel esencial en la motivación a los niños en seguir las tareas de rehabilitación, ofreciendo a los menores una rehabilitación divertida e inmersiva. Esto es especialmente importante debido a la prolongada duración de los tratamientos de rehabilitación, como suele ser habitual en los pacientes con parálisis cerebral.

En el marco del TFG se diseñará e implementará un entorno virtual acorde con la población infantil. Tras esto se establece una comunicación bidireccional entre el entorno de realidad virtual y la plataforma robótica CPWalker, con el fin de emular sus movimientos en el entorno virtual. Además, desde el entorno de realidad virtual se puede enviar órdenes al CPWalker a través del control de los distintos módulos de accionamiento de este. Esto permite establecer una sesión de terapia personalizada con unos parámetros específicos enviando comandos al CPWalker y recibiendo los datos de éste, manteniendo una sincronización en tiempo real durante toda la terapia.

### **PALABRAS CLAVE:**

Realidad Virtual, terapia asistida por robots, neurorrehabilitación

## SUMMARY

The aim of this BSc Thesis is to design and implement a Virtual Reality (VR) environment as a framework to study its role in the field of neurorehabilitation, especially in children with cerebral palsy.

For this purpose, a virtual environment has been integrated into the CPWalker robotic platform designed by the CSIC (Centro Superior de Investigaciones Científicas). It is the first walker with dynamic support of body weight and active gait designed to be applied in real environments. The incorporation of virtual reality into robotic-assisted therapy plays an essential role in motivating children to follow rehabilitation tasks, offering children a fun and immersive rehabilitation. This is especially important due to the long duration of rehabilitation treatments, as is often the case with cerebral palsy patients.

Within the framework of the BSc Thesis, a virtual environment will be designed and implemented according to the child population. After this, a bidirectional communication will be established between the virtual reality environment and the CPWalker robotic platform, to emulate their movements in the virtual environment. In addition, the virtual reality environment can send commands to the CPWalker by controlling the different drive modules of the CPWalker. This will allow us to establish a personalised therapy session with specific parameters by sending commands to the CPWalker and receiving data from it, maintaining real-time synchronisation throughout the therapy.

### **KEYWORDS:**

Virtual Reality, robot-assisted therapy, neurorehabilitation.

## **Agradecimientos**

Querría agradecer a mis tutores Álvaro Gutiérrez y Eduardo Rocón, por la ayuda y los consejos recibidos en la realización del presente Trabajo Fin de Grado.

Muchas gracias.

A todos los compañeros de CSIC-CAR, especialmente a Pablo Romero, cuya ayuda ha sido imprescindible.

Muchas gracias.

A mi familia, mis amigos, y a Javi, por su apoyo y ánimo.

Muchas gracias.



# ÍNDICE DEL CONTENIDO

<b>I. INTRODUCCIÓN .....</b>	<b>1</b>
1.1 INTRODUCCIÓN .....	1
1.2 PARÁLISIS CEREBRAL INFANTIL .....	1
1.3 TIPOS DE PARÁLISIS CEREBRAL Y SU TRATAMIENTO .....	1
2. NEURORREHABILITACIÓN Y NEUROPLASTICIDAD.....	5
2.1 LA ROBÓTICA .....	5
2.1.1 LA PLATAFORMA ROBÓTICA CPWALKER .....	6
2.2 REALIDAD VIRTUAL (RV).....	7
<b>II. OBJETIVOS .....</b>	<b>9</b>
<b>III. DISEÑO E IMPLEMENTACIÓN.....</b>	<b>10</b>
1. ESTRUCTURA DEL TRABAJO.....	10
2. DISEÑO DEL ENTORNO VIRTUAL .....	11
2.1 EL ENTORNO VIRTUAL Y EL AVATAR.....	13
3. ARQUITECTURA DE COMUNICACIÓN.....	17
3.1 ARQUITECTURA DE CONTROL.....	17
3.2 RECEPCIÓN Y DECODIFICACIÓN DE DATOS .....	23
3.3 LA CINEMÁTICA EN UNITY .....	28
4. CONCLUSIONES .....	32
<b>IV. RESULTADOS.....</b>	<b>33</b>
1. DEFINICIÓN DE LA SIMULACIÓN.....	33
2. ADQUISICIÓN Y ANÁLISIS DE LOS DATOS .....	36
<b>V. CONCLUSIONES Y LÍNEAS FUTURAS .....</b>	<b>40</b>
1. CONCLUSIONES .....	40
1.1 OTROS ASPECTOS INTERESANTES DEL DESARROLLO .....	40
2. LÍNEAS FUTURAS .....	42
<b>VI. BIBLIOGRAFÍA .....</b>	<b>44</b>
<b>ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES .....</b>	<b>47</b>
1. IMPACTO MEDIOAMBIENTAL.....	47
2. IMPACTO ÉTICO, SOCIAL Y ECONÓMICO .....	47
<b>ANEXO B: PRESUPUESTO ECONÓMICO .....</b>	<b>49</b>

## LISTA DE ACRÓNIMOS

**CAN:** Controller Area Network

**CSIC:** Centro Superior de Investigaciones Científicas

**GMFCS:** Gross Motor Function Classification System

**HMD:** Head Mounted Display

**IDE:** Integrated Development Environment Computing

**ODS:** Objetivos de Desarrollo Sostenible

**ONU:** Organización de las Naciones Unidas

**OSI:** Open Systems Interconnection

**PBWS:** Partial Body Weight Support

**PCI:** Parálisis Cerebral Infantil

**RA:** Realidad Aumentada

**ROM:** Range Of Motion

**RV:** Realidad Virtual

**UDP:** User Datagram Protocol

**WLAN:** Wireless Local Area Network

**2D:** Two Dimensional

**3D:** Three Dimensional



# I. INTRODUCCIÓN

## 1.1 INTRODUCCIÓN

Los niños son un grupo de población con características diferenciales frente al resto de edades, donde el cerebro y el cuerpo se encuentran en pleno desarrollo y la plasticidad neuronal es máxima. La intervención de la robótica y las tecnologías de la realidad virtual en su rehabilitación, en comparación con las terapias convencionales, aportan beneficios a base de estimular su cerebro, ya sea aumentando el número y la intensidad de los ejercicios o la motivación en participar y cooperar [1].

## 1.2 PARÁLISIS CEREBRAL INFANTIL

La Parálisis Cerebral Infantil (PCI) engloba una serie de trastornos motores, caracterizados por un grupo de alteraciones no progresivas del movimiento y postura que limitan la actividad, debida a una lesión del cerebro causada durante el desarrollo cerebral del feto o en la gestación, el parto o los primeros años de vida. Si bien este tipo de lesión es estable, la manifestación de los trastornos motores puede variar ya que dependen de la extensión y la localización de la lesión, así como la capacidad del cerebro a adaptarse a ella [1].

La PCI es en la actualidad la causa más frecuente de discapacidad motora en la infancia, a pesar de los avances en el cuidado de la embarazada y la medicina prenatal, es un problema de salud que mantiene una incidencia a nivel mundial alrededor de 2 o 3 por cada 1000 bebés nacidos vivos, de acuerdo con los estudios referentes realizados recientemente [2]. Los pacientes con parálisis cerebral presentan limitaciones en la postura y movilidad, que reducen la participación de las actividades físicas y sociales. Además de otras condiciones coexistentes como sensoriales o cognitivas que comúnmente ocurren en la PCI [3].

## 1.3 TIPOS DE PARÁLISIS CEREBRAL Y SU TRATAMIENTO

La parálisis cerebral incluye grupos heterogéneos de trastornos, por lo que es difícil de clasificar el tipo de PCI que tiene un paciente. Asimismo, existen diferentes criterios de clasificación: según el lugar de lesión en el cerebro; en función de la parte del cuerpo a la que afecta o atendiendo el nivel de severidad.

La clasificación según el sitio de la lesión en el cerebro (ver la [Figura 1.1](#) [4]):

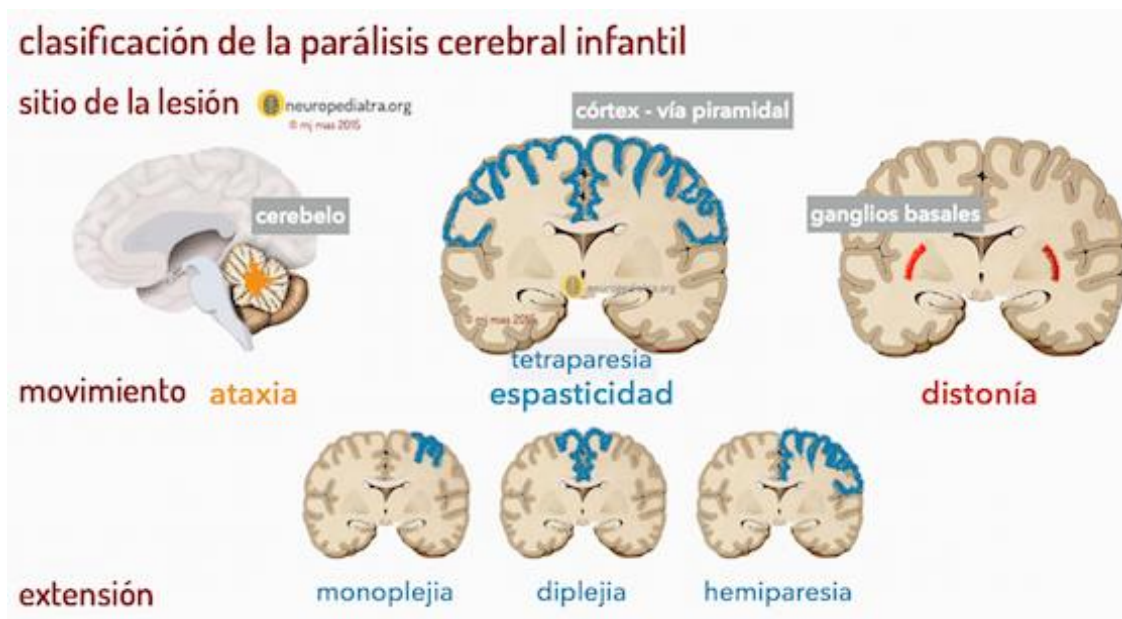
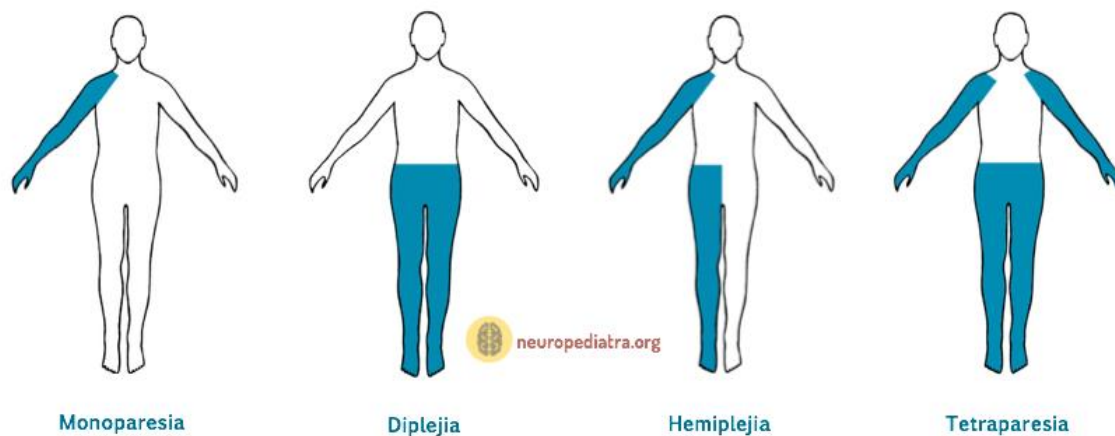


Figura 1.1. Clasificación de la parálisis cerebral según el sitio de la lesión [5].

- La **parálisis cerebral espástica** es la más frecuente, se da entre el 70-80% de niños con PCI aproximadamente. Este tipo de parálisis supone una rigidez muscular, las extremidades afectadas tienen el músculo en contracción constante y el músculo antagonista muy débil. Esto produce un desequilibrio de fuerzas, causando movimientos abruptos o repetidos. Este tipo de parálisis se produce frecuentemente por el daño en células nerviosas de la corteza cerebral.
- La **parálisis cerebral discinética** es la menos frecuente, únicamente un 10-15% de pacientes con parálisis cerebral lo padecen. Se caracteriza por un cambio brusco del tono muscular, debido a las contracciones involuntarias de los músculos tanto en reposo como al iniciar el movimiento. Esta alteración se corresponde con una lesión en los ganglios basales, región que desempeñan un papel importante en el control de la postura y el movimiento voluntario. Los pacientes con este tipo de parálisis cerebral tienen problemas para controlar los movimientos de sus extremidades, lo que les dificulta caminar o sentarse.
- La **parálisis cerebral atáxica** se relaciona con problemas de equilibrio y falta de coordinación, funciones que se corresponden con el cerebelo. De modo que las personas que la padecen presentan inestabilidad al caminar y dificultades de controlar los movimientos finos, como escribir.
- La **parálisis cerebral mixta** se produce cuando hay lesiones en diferentes estructuras del cerebro, por lo que las características de los diferentes tipos de parálisis no se manifestarán en sus formas puras, sino que existirán combinaciones.

La clasificación según los criterios topográficos (ver [Figura 1.2](#)):

- **Monoparesia**, afecta sólo a un miembro del cuerpo, normalmente el brazo.
- **Diplejía**, afecta a las dos piernas, estando los brazos nada o ligeramente afectados.
- **Hemiplejia**, se produce cuando sólo se lesiona el lado izquierdo o derecho del cuerpo.
- **Tetraplejia**, están afectados los dos brazos y las dos piernas.



*Figura 1.2. Clasificación de la parálisis cerebral espástica según la extensión anatómica [5].*

Por último, la clasificación según “Gross Motor Function Classification System” (GMFCS) es la escala más utilizada para categorizar el grado de afección en las funciones motoras de los niños con parálisis cerebral y la necesidad de los dispositivos de asistencia y/o de movilidad con ruedas [6]. Consiste en una clasificación de cinco niveles, como se puede observar en la [Figura 1.3](#).

- **Nivel I:** Los niños pueden correr y saltar, pero tienen limitaciones en cuanto a velocidad, equilibrio y la coordinación.
- **Nivel II:** Los niños pueden subir escaleras agarrados a una barandilla, pero tienen limitaciones para caminar por superficies irregulares o a distancias largas.
- **Nivel III:** Los niños pueden caminar utilizando un dispositivo de movilidad manual.
- **Nivel IV:** Los niños pueden lograr la auto movilidad sólo si emplean andadores o sillas de ruedas eléctrica.
- **Nivel V:** Los niños tienen graves limitaciones en el control de la cabeza y los movimientos de las extremidades. Necesitan ser transportados en una silla de ruedas.

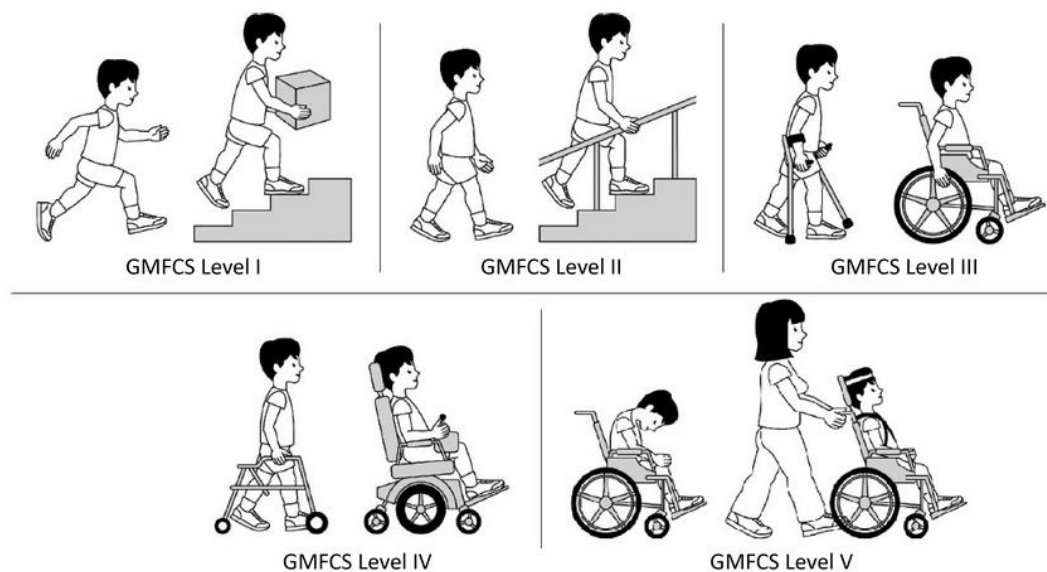


Figura 1.3. Clasificación según la escala Gross Motor Function Classification System (GMFCS) [7].

Los tipos de tratamiento más comunes para la parálisis cerebral incluyen [8,9]:

- **Fisioterapia y rehabilitación.** Los niños con parálisis cerebral generalmente comienzan estos tratamientos al año o poco después de ser diagnosticados. La fisioterapia es una de las partes más importantes del tratamiento. Incluye cualquier ejercicio o actividad que pueda mantener o mejorar la fuerza, el equilibrio y el movimiento de los músculos. Un fisioterapeuta ayudará al niño a desarrollar habilidades como sentarse, caminar y usar una silla de ruedas. Otros tipos de tratamiento incluyen terapia ocupacional, terapia recreativa y terapia del habla.
- **Instrumentos ortopédicos.** Se pueden colocar férulas y vendajes para caminar en las extremidades afectadas para mejorar el equilibrio con el movimiento. Otros dispositivos que pueden ayudar con el movimiento y la postura incluyen sillas de ruedas, andadores y patinetes eléctricos.
- **Equipos y tecnología auxiliares.** Estos incluyen dispositivos especiales de comunicación por computadora, zapatos abrochados con velcro y muletas para hacer la vida cotidiana más cómoda.
- **Medicación y cirugía.** Algunos medicamentos ayudan a relajar los músculos rígidos e irritados y a reducir los movimientos irregulares. Sin embargo, algunos de estos medicamentos también pueden afectar negativamente el desarrollo del cerebro. Es posible que se requiera cirugía si los síntomas son graves. Por ejemplo, estirar los músculos que se han vuelto rígidos.

Sin embargo, no todas las terapias son apropiadas para todas las personas con parálisis cerebral. Existe un consenso en cuanto a que el tratamiento y la colaboración de otros

especialistas debe ser multidisciplinaria para la valoración y atención integral del niño con parálisis cerebral. Además, debe ser individualizado, en función de la situación de cada paciente. Una atención temprana durante los primeros años y un tratamiento de mantenimiento posterior.

## 2. NEURORREHABILITACIÓN Y NEUROPLASTICIDAD

La neurorrehabilitación es un “proceso activo por medio del cual los individuos con alguna lesión o enfermedad pueden alcanzar la recuperación integral óptima posible, que les permite el desarrollo físico, mental y social de la mejor forma, para integrarse a su medio ambiente de la manera más apropiada”. Por otro lado, la neuroplasticidad o la plasticidad cerebral, es la habilidad del cerebro de cambiar tanto su estructura física como su organización funcional, en respuesta a una serie de estímulos, y entrenamientos cognitivos, que favorecen nuevas interconexiones neuronales, y que refuerzan las conexiones ya existentes [10].

La terapia debe basarse en tareas repetitivas orientadas a objetivos y con compromiso cognitivo puede ser de particular interés en pediatría donde la neuroplasticidad se percibe al máximo, ya que permiten que los cambios inducidos por el ejercicio se detecten y fijen en la corteza motora, los ganglios basales, el cerebelo y la columna vertebral [11]. Varios grupos informan de una mejoría constante en la marcha en niños con parálisis cerebral. Además, los padres y los pacientes informaron una mejor calidad de vida [12].

Bajo este contexto, se van a estudiar los distintos enfoques de la neurorrehabilitación para lograr mejoras significativas en los resultados del tratamiento en niños con parálisis cerebral. Dentro de las múltiples opciones existentes, se distinguen dos líneas principales, por un lado: la intervención sin ningún equipamiento tecnológico, como la terapia de movimiento inducido por restricción del miembro sano o CIMT (Constraint Induced Movement Therapy) empleando manoplas especializadas, cabestrillos, vendas triangulares o guantes [13]; y por otro, usando equipos tecnológicos, como la robótica o las tecnologías de la realidad virtual, que son los que se van a tratar en este trabajo.

### 2.1 LA ROBÓTICA

La terapia asistida por robots tiene el potencial de aumentar la fuerza funcional y mejorar los movimientos aislados, dado que puede ofrecer una asistencia constante además de establecer la intensidad y dificultad de acuerdo con la mejora del paciente.

Los dispositivos robóticos de asistencia se dividen en dos grandes grupos: los dispositivos alternativos y los de potenciación. Son seleccionados en función del grado de discapacidad del usuario, asimismo, las personas con movilidad reducida suelen emplear dispositivos



augmentativos ya que estos les permiten aprovechar su fuerza residual. Los andadores o exoesqueletos son ejemplos de este tipo de dispositivos y ayudan al usuario a estar de pie, en equilibrio o caminar [14].

Los robots de tipo exoesqueleto son dispositivos mecatrónicos que sirven de apoyo a las extremidades débiles e incluso pueden llegar a sustituir la función de alguno de los miembros afectados. Los segmentos y articulaciones del exoesqueleto se corresponden en cierta medida con los del cuerpo humano y el sistema está acoplado externamente a la persona. En las aplicaciones de rehabilitación, los exoesqueletos deben ser capaces de replicar, con un paciente, los movimientos realizados con un terapeuta durante el tratamiento. Asimismo, deben permitir a las personas con movilidad reducida realizar las actividades diarias [15].

Adicionalmente, puede haber sensores acoplados al exoesqueleto que permitan evaluar la fuerza y los movimientos del paciente, lo que proporcionaría al terapeuta información cuantitativa sobre el proceso de recuperación y rehabilitación del paciente. Por lo tanto, el exoesqueleto podría actuar como una herramienta para la medición del rendimiento y la evolución del tratamiento.

### 2.1.1 LA PLATAFORMA ROBÓTICA CPWALKER

El CPWalker [16,17] es una plataforma robótica diseñada por el CSIC (Centro Superior de Investigaciones Científicas) para la rehabilitación de marcha de pacientes con parálisis cerebral, cuyo objetivo es promover la incorporación temprana de los pacientes con PCI a la terapia de rehabilitación, ayudando a aumentar la intensidad y frecuencias de los ejercicios de rehabilitación, lo que permitirá el mantenimiento de los métodos terapéuticos a diario y lograr así mejorías en los resultados del tratamiento.

Esta plataforma robótica está compuesta de dos estructuras principales, un andador inteligente para "ambulación" y un exoesqueleto para el control de las articulaciones, permitiendo así terapias de rehabilitación basada en la robótica [16]. La estructura resiste un total de 80 kg (el exoesqueleto y el paciente) aproximadamente y está subdividido en los siguientes módulos:

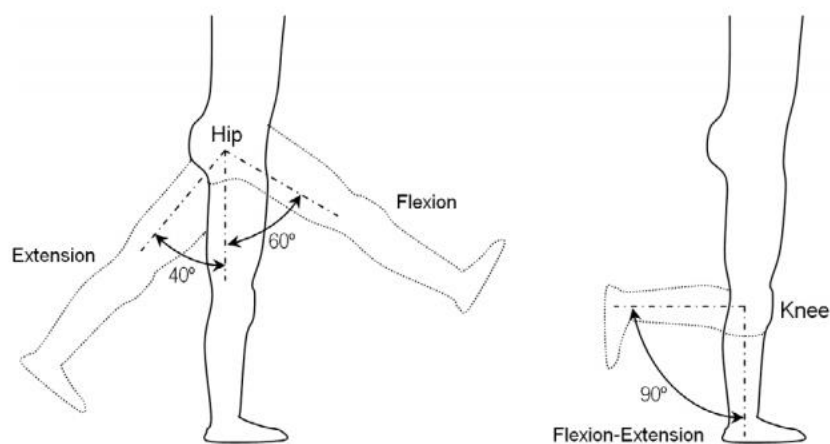
- **Sistema de tracción:** este módulo está situado en las ruedas traseras y se encarga de generar el movimiento de la plataforma. Está compuesto por dos motorreductores independientes acoplados en cada rueda trasera y otros dos encoders para el control de la velocidad de traslación.

- **Sistema de soporte de peso corporal parcial:** este módulo está diseñado para el control de la descarga del peso corporal de los pacientes. La descarga parcial del peso del usuario durante la marcha ayuda a mejorar los resultados de la rehabilitación. Está compuesto por un actuador lineal eléctrico que controla el peso del paciente; un potenciómetro para el control de la altura de la cadera y una célula de carga para medir el peso soportado en ese momento. Este módulo está diseñado para adaptar la plataforma a las diferentes tallas de los

pacientes, ajustando la articulación de la cadera del exoesqueleto a diferentes alturas. Se usa también para ir subiendo y bajando la cadera durante la trayectoria, igual que en una marcha sana.

- **El exoesqueleto del CPWalker** es un robot con una configuración cinemática similar a las extremidades inferiores humanas, compuesto por cuatro articulaciones activas (rodillas y caderas derecha e izquierda) y dos pasivas (tobillos).

Las dimensiones de la estructura también pueden ajustarse a los diferentes tamaños antropomórficos de los pacientes. El movimiento mecánico está limitado a algunos rangos para ajustarse a la marcha sobre el suelo y evitar que el paciente sufra caídas. El rango de ángulos de las caderas es de  $[60^\circ, -40^\circ]$ ; y el rango de las rodillas es entre  $[90^\circ, 0^\circ]$ , como se puede observar en la [Figura 1.4](#).



*Figura 1.4. El rango de movimiento de diferentes articulaciones del exoesqueleto del CPWalker.*

## 2.2 REALIDAD VIRTUAL (RV)

La realidad virtual consiste en representar escenas o imágenes de objetos mediante un sistema informático, produciendo una sensación de existencia real. Es decir, se trata de una simulación de un entorno real con ordenador y se caracteriza por la inmersión e interacción.

Los niños con parálisis cerebral pueden tener menos oportunidades para jugar libremente debido a las limitaciones de movilidad y habilidad manual. Sin la oportunidad de jugar de forma voluntaria y espontánea, los niños pueden desarrollar una impotencia en el aprendizaje y asumir que son incapaces de realizar una tarea. Para mejorar la eficacia de la terapia de ejercicios y la mecanoterapia en niños con parálisis cerebral, es importante lograr un papel más activo del paciente en el entrenamiento. Como han demostrado los estudios, las habilidades motoras se adquieren de forma más eficiente y se mantienen durante más tiempo cuando el paciente participa activamente en la generación de un patrón de movimientos en vez de cuando sólo se realizan movimientos pasivos [18].

El uso de la realidad virtual en la rehabilitación pediátrica se basa en la propiedad única de brindar oportunidades para participar en actividades divertidas, motivadoras, y que, al mismo tiempo, sean agradables y seguras para el aprendizaje activo. Su importancia se relaciona con la motivación para repetir una tarea varias veces y, es aún más importante para los niños que no siguen un programa de ejercicio convencional porque consideran que esos ejercicios son menos interesantes.

Los dispositivos de RV se pueden clasificar en función del tipo de pantalla de visualización y su contenido gráfico. Las pantallas de visualización pueden ser ordenadores (no inmersivos), pantallas panorámicas (semi inmersivos), dispositivos sobre la cabeza (HMD por sus siglas en inglés) e incluso salas de realidad virtual (inmersivos). En función de su contenido gráfico, la realidad virtual puede presentarse en 2D o 3D [19]. La inmersión permite al usuario sentir su presencia dentro del mundo virtual, y vendrá dado por la percepción objetiva del sujeto en función de los estímulos que el sistema le aporte, en términos de la cantidad y calidad. Asimismo, la interacción permite al usuario interactuar con los objetos del entorno virtual, en lugar de ser simplemente una representación gráfica. Además, es capaz de proporcionar realimentación sensorial al paciente; ésta puede venir dada de forma visual, auditiva o táctil, e incluso a través del movimiento. Esta realimentación ayuda al paciente a ajustar sus movimientos, facilitando así el aprendizaje a través de determinadas vías neuronales [20].

Sin embargo, una de las principales limitaciones es la falta de estudios transversales donde el tamaño de la muestra sea lo suficientemente grande. Los investigadores han de verificar la contribución añadida de la RV a los tratamientos existentes y empíricamente validados. La evidencia positiva de grandes estudios comparativos respaldaría el uso de la RV como herramienta de refuerzo. Además, aunque los resultados de Brutsch et al. [21] sugieren que la RV desempeña un papel en el aumento de la participación de los pacientes, los estudios actuales aún no han determinado el tipo o el alcance del efecto terapéutico de la RV.

Por lo tanto, aunque muchos estudios sugieren que la realidad virtual mejora la eficacia de los programas existentes al incrementar la motivación y la participación, todavía no se ha medido su calidad y cantidad.

## II. OBJETIVOS

Dadas las ventajas de la robótica y las tecnologías de la realidad virtual en el campo de la neurorrehabilitación de pacientes con parálisis cerebral, se ha propuesto diseñar un entorno virtual que simule una sesión de terapia con los parámetros que se sincronizarán con el robot CPWalker. Ambas tecnologías estarán integradas para una mayor inmersión del paciente, permitiendo variar parámetros de la terapia desde el entorno virtual, algunos ejemplos son: número de pasos, el rango de movimiento, la velocidad de tracción o el nivel de impedancia. Para ello, se ha establecido los siguientes objetivos:

- Diseñar un entorno de Realidad Virtual adaptada a la población infantil y la rehabilitación de marcha asistida por el CPWalker.
- Desarrollar y validar el protocolo de comunicación entre el robot CPWalker y la RV.
- Aplicar los datos de las articulaciones del CPWalker en el avatar del entorno virtual para lograr la sincronización en el movimiento.
- Diseñar una sesión de terapia mediante la definición de un conjunto de parámetros que el CPWalker debe seguir.

### III. DISEÑO E IMPLEMENTACIÓN

Este capítulo está dividido en varias partes, primero se va a exponer una visión general del proyecto con sus características más importantes. Seguidamente se va a realizar un desglose del entorno desarrollado, donde se analizarán sus principales componentes, desde la implementación del escenario virtual con Unity3D, hasta la interconexión entre este y el CPWalker. Tras esto, se explicará la decodificación de los datos recibidos y la manera de aplicar estos datos en el avatar para conseguir la sincronización del movimiento con el CPWalker. Por último, se expondrán los diferentes problemas que han surgido durante su programación y las decisiones que se han tomado para solventarlos.

#### 1. ESTRUCTURA DEL TRABAJO

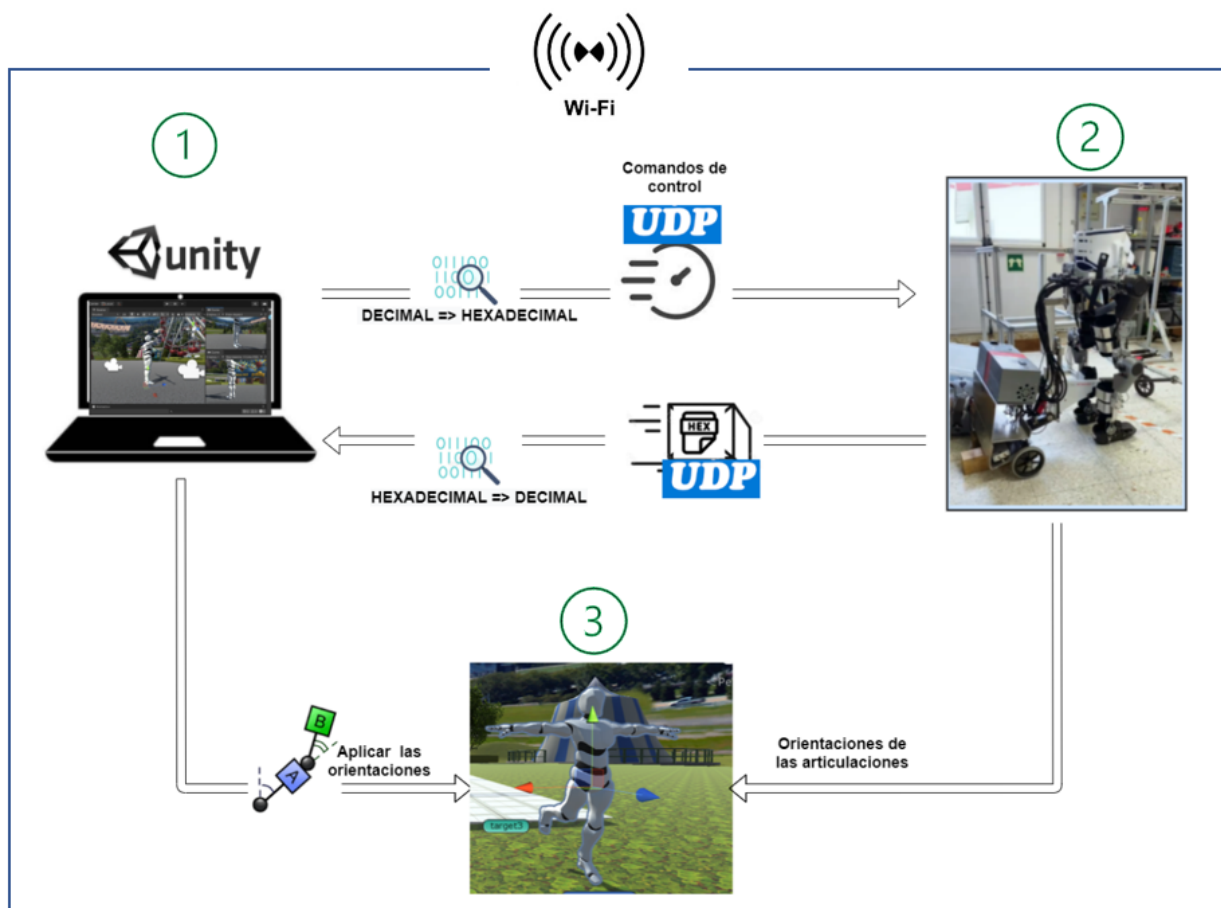


Figura 2.1. Esquema general del funcionamiento del sistema.

Como podemos observar en la [Figura 2.1](#), la estructura de este sistema consta de tres componentes principales y unos procesos asociados a cada uno de ellos, que se repiten mientras el sistema esté ejecutándose.

El funcionamiento en líneas generales es como se describe a continuación:

1. Se establece la conexión entre el entorno de la realidad virtual (Unity) con el CPWalker a través de una red Wifi.
2. Desde Unity se inicia la ejecución, enviando una serie de órdenes de control al CPWalker, éste se moverá hasta completar la tarea que se le ha asignado.
3. Mientras que el CPWalker esté en acción, emitirá paquetes de mensajes que contienen la información sobre la orientación de las articulaciones.
4. Desde el entorno virtual se leen esos datos, se decodifican y se extraen los datos de interés (las orientaciones de las articulaciones en cada instante del tiempo).
5. Se aplican estos datos a cada una de las articulaciones del avatar emulando el movimiento del CPWalker, manteniendo una sincronización del movimiento en tiempo real.

## 2. DISEÑO DEL ENTORNO VIRTUAL

Hemos seleccionado la plataforma Unity para el desarrollo del entorno de la realidad virtual. Unity es un motor gráfico multiplataforma creado por *Unity Technologies*, se usa principalmente para desarrollar videos y simulaciones tanto en 3D como en 2D para ordenadores, consolas y dispositivos móviles [22].

La interfaz de Unity3D consta de varias pestañas y una barra de herramientas que se utilizan para crear todos los subsistemas de la simulación virtual. Los principales componentes en la interfaz de Unity, de izquierda a derecha en la *Figura 2.2* son las siguientes:

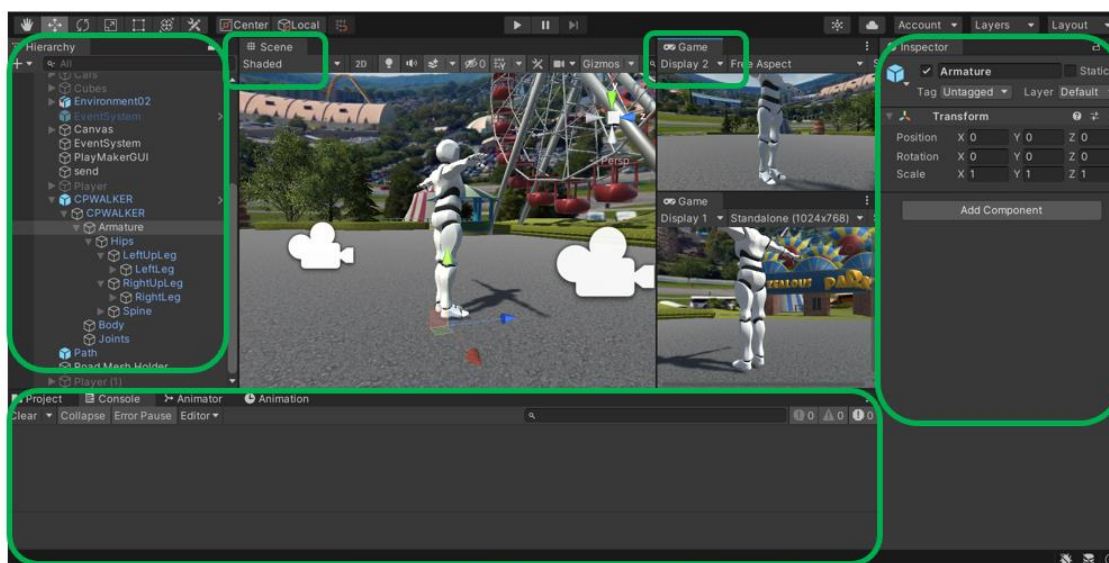


Figura 2.2. Las distintas pestañas de Unity3D.

1. **Hierarchy (vista de jerarquía):** contiene todos los objetos en la escena actual.
2. **Scene (vista de escena):** es el área de construcción de Unity donde se construye visualmente cada escena. Los componentes esenciales de una escena son los *GameObjects* u objetos. Al crear una escena por primera vez, por defecto viene la cámara y la luz de tipo direccional, pero son configurables según el diseño.
3. **Game (vista de juego):** En esta vista se obtiene una previsualización de la escena creada previamente.
4. **Inspector:** La vista de inspector sirve tanto para mostrar propiedades de los objetos como agregar características a los mismos.
5. **Project o Proyecto:** contiene todas las librerías, objetos prefabricados y otros materiales que pueden ser utilizados durante el desarrollo del proyecto.

Como se observa en la [Figura 2.3](#), cada elemento en la escena es un *GameObject* u objeto, que posee ciertas propiedades (texturas, mallas, materiales...) y comportamientos (definidos por desarrollador a través de los *Scripts*). Algunos de sus atributos pueden ser modificados tanto por código como desde la pestaña del editor. Todos los componentes pueden ser activados o desactivados excepto el *Transform* que viene por defecto en todos los *Gameobjects* y contiene la posición, rotación y escala de cada objeto. Los *GameObjects* pueden ser anidados unos con otros estableciendo una jerarquía de padre e hijo, generando así una dependencia que resulta ser interesante e importante, especialmente en la configuración del avatar.

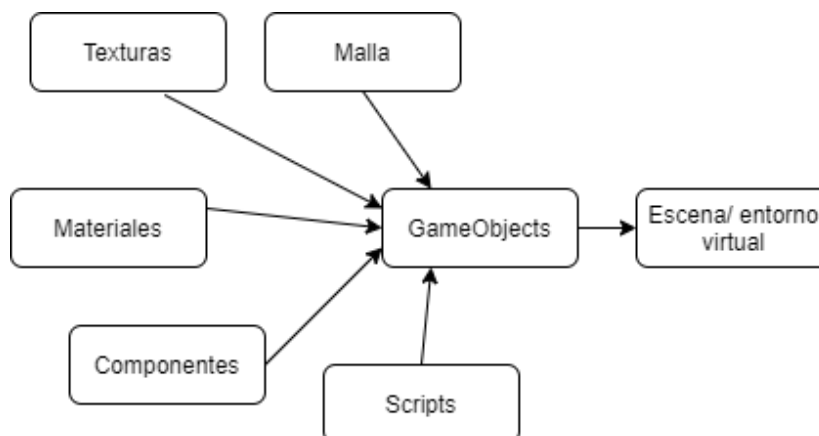


Figura 2.3. Distintos componentes de los *GameObjects* del entorno virtual.

Los objetos de una escena se crean al cargar y se destruyen al cambiar a otra, por lo que para conservar alguno de los objetos entre escenas debe ser especificado por código.

Para implementar el comportamiento de cada objeto se ha empleado el lenguaje de programación C#. Es un lenguaje que se deriva de C y C++ y fue creado por Microsoft como parte de las plataformas .NET. A través de los scripts se puede acceder tanto a las librerías propias de .NET como a las clases y funciones definidas en Unity3D haciendo llamadas a UnityEngine. Los scripts deben heredar de la clase *MonoBehaviour* que permite el acceso a varios métodos utilizados en este proyecto, entre los cuales destacan *Start()*, que se ejecuta



cuando el script se lanza por primera vez en la escena, y *Update()*, que se ejecuta continuamente una vez por fotograma mientras el script está activo.

Aprovechando la versatilidad que ofrece Unity a la hora de diseñar los entornos virtuales y los mecanismos de interacción, se pueden crear escenarios adaptados a las necesidades de cada paciente, logrando así un tratamiento lo más personalizado posible.

## 2.1 EL ENTORNO VIRTUAL Y EL AVATAR

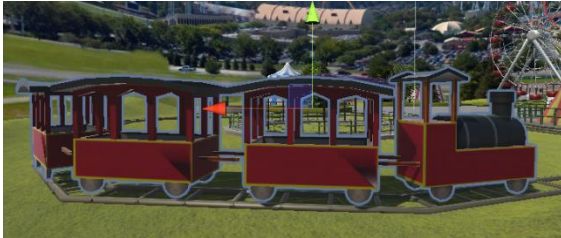
Como se puede apreciar en las *Figuras 2.4 a 2.10*, el entorno virtual pretende simular un parque temático en el que hay diversas atracciones. Se pretende construir un escenario divertido y entretenido, especialmente para los niños con movilidad reducida.

Las referencias en el mundo virtual siguen la distribución de los ejes de coordenadas cartesianas en el que el eje X corresponde al movimiento de derecha a izquierda; el eje Y permite subir y bajar con el efector final; y el eje Z se relaciona con la profundidad que permite movimientos de avance y retroceso.

La creación del entorno virtual se basa en la agregación de los distintos elementos decorativos a modo de *prefabs*. Los *prefabs* son un tipo objetos prefabricados con distintas propiedades, como es el mallado, la textura, el color e incluso la animación. Estos recursos pueden ser encontrados en el *Asset Store* de Unity, tras la descarga se importan y se guardan en la ventana de *Project*, otros ejemplos de estos recursos son paquetes de vegetación, de cielos, de personajes, etc.

Se han creado varias cámaras móviles con enfoques desde distintos puntos de vista. La más importante es la de primera persona, que permite al usuario observar la escena como si fuera a través de sus ojos y la del observador para una mejor visualización de lo que está ocurriendo en el escenario y, sobre todo, observar el movimiento del avatar. Por último, se han agregado los audios de ambiente característicos de un parque de atracciones para una mayor inmersión del usuario.





*Figura 2.4. Un tren que gira en su carril.*



*Figura 2.5. Una noria con animación.*



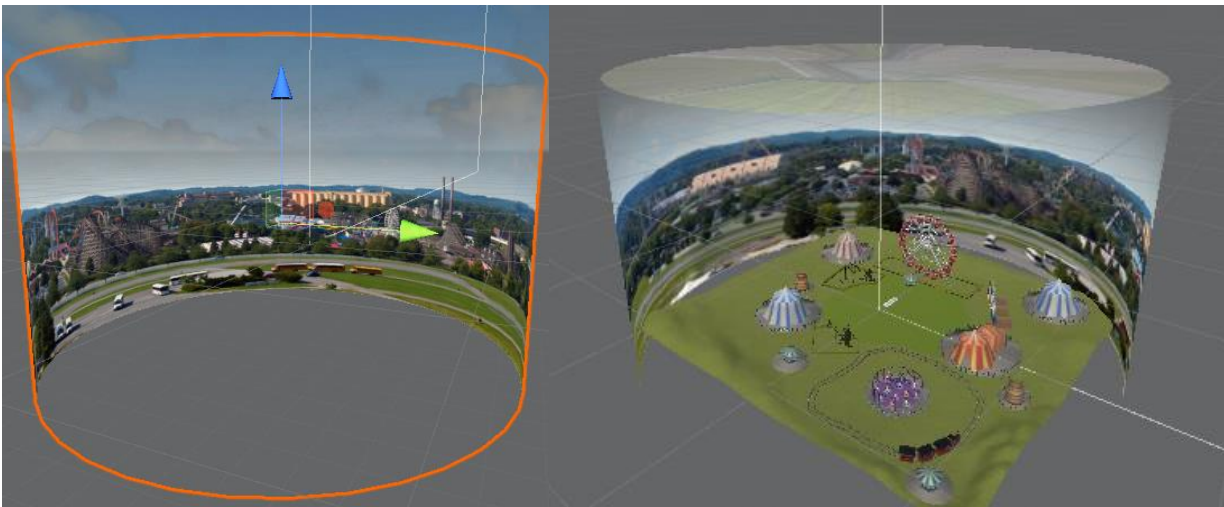
*Figura 2.6. Atracción de tazas giratorias.*



*Figura 2.7. Un árbol de decoración.*



*Figura 2.8. La escena con los elementos decorativos incorporados.*



*Figura 2.9. Un fondo panorámico para una mayor inmersión.*

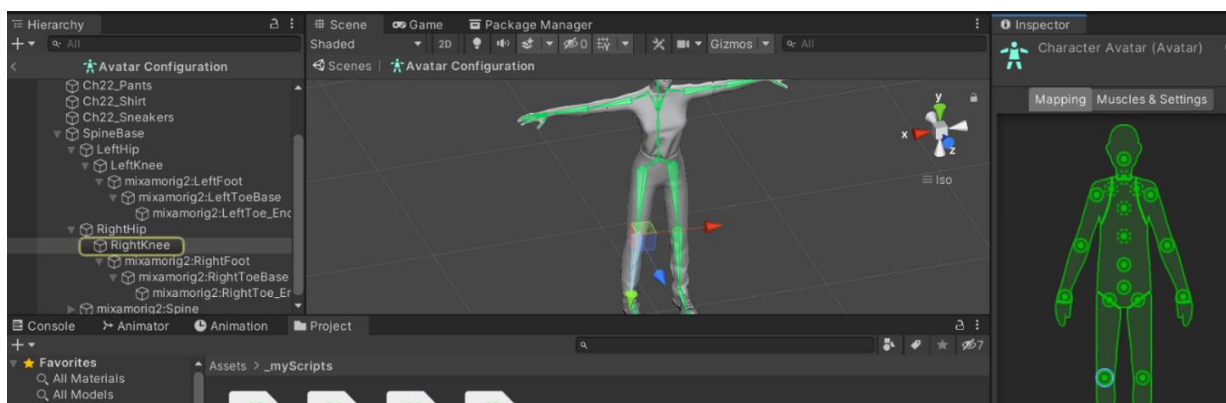


*Figura 2.10. Escenario final vista desde los ojos del avatar*

Con el fin de dar mayor realismo a la simulación se ha elegido el avatar de tipo humanoide. Para su correcta configuración, se ha de seleccionar el "Rig", también conocido como esqueleto. Se trata de un modelo simplificado de un esqueleto humano con extremidades y articulaciones definidas, formando una jerarquía con múltiples niveles de relaciones padre-hijo. Es importante esta configuración para el correcto movimiento del avatar, puesto que, en caso contrario, cada objeto tomará el centro del mundo como referencia y se moverán con respecto a este.

Cualquier nodo puede tener varios hijos y los objetos hijos también pueden tener hijos propios y así sucesivamente, pero cada nodo tiene sólo un padre. El objeto en la cima de la jerarquía se conoce como la raíz, en este modelo concreto, es la pelvis y será la referencia del resto de objetos de la jerarquía. Asimismo, la pelvis tiene como hijos los dos muslos, cada uno de los muslos con sus piernas correspondientes, y cada uno de los pies dentro de cada pierna. Normalmente dicha configuración ya viene dada al importar el modelo del avatar, para la consulta y modificación se ha de realizar en la pestaña de *Hierarchy* (ver [Figura 2.11](#)).

Posteriormente se deja a Unity finalizar con el mapeo de los huesos y las articulaciones del avatar (ver [Figura 2.11](#)). Se puede observar que el modelo está en posición T porque es la posición predefinida en Unity. El mapeo del esqueleto indica cuáles son los huesos imprescindibles (círculos sólidos) y cuáles de los huesos son opcionales (círculos punteados). Para los huesos opcionales, Unity puede interpolar sus movimientos automáticamente. Asimismo, cada nodo se referencia con un nombre y contiene una posición, una rotación y restricciones sobre la forma en que el hueso puede girar y rotar. Aprovechando este hecho, seremos capaces de replicar el movimiento del paciente transmitiendo la orientación de las articulaciones reales a las virtuales.



*Figura 2.11. La ventana del mapeo de las articulaciones del avatar humanoide.*

Con esto, el avatar ya cumple los requisitos básicos para moverse emulando los movimientos del CPWalker y, por tanto, los del paciente. El siguiente paso es establecer la interconexión entre Unity y el CPWalker para el intercambio de información.

### 3. ARQUITECTURA DE COMUNICACIÓN

Esta sección describe los métodos y pasos que se ha seguido para lograr la interconexión entre el CPWalker y Unity3D. Se trata de una comunicación bidireccional y está dividida en dos subapartados. En primer lugar, se pretende mandar comandos de control al CPWalker y, en segundo lugar, recibir los datos de salida de este y finalmente, emplear la información recibida del CPWalker para lograr una sincronización con el entorno virtual.

#### 3.1 ARQUITECTURA DE CONTROL

Para lograr los objetivos descritos, se debe entender cómo es la arquitectura de control del CPWalker y el protocolo de comunicación que se debe emplear.

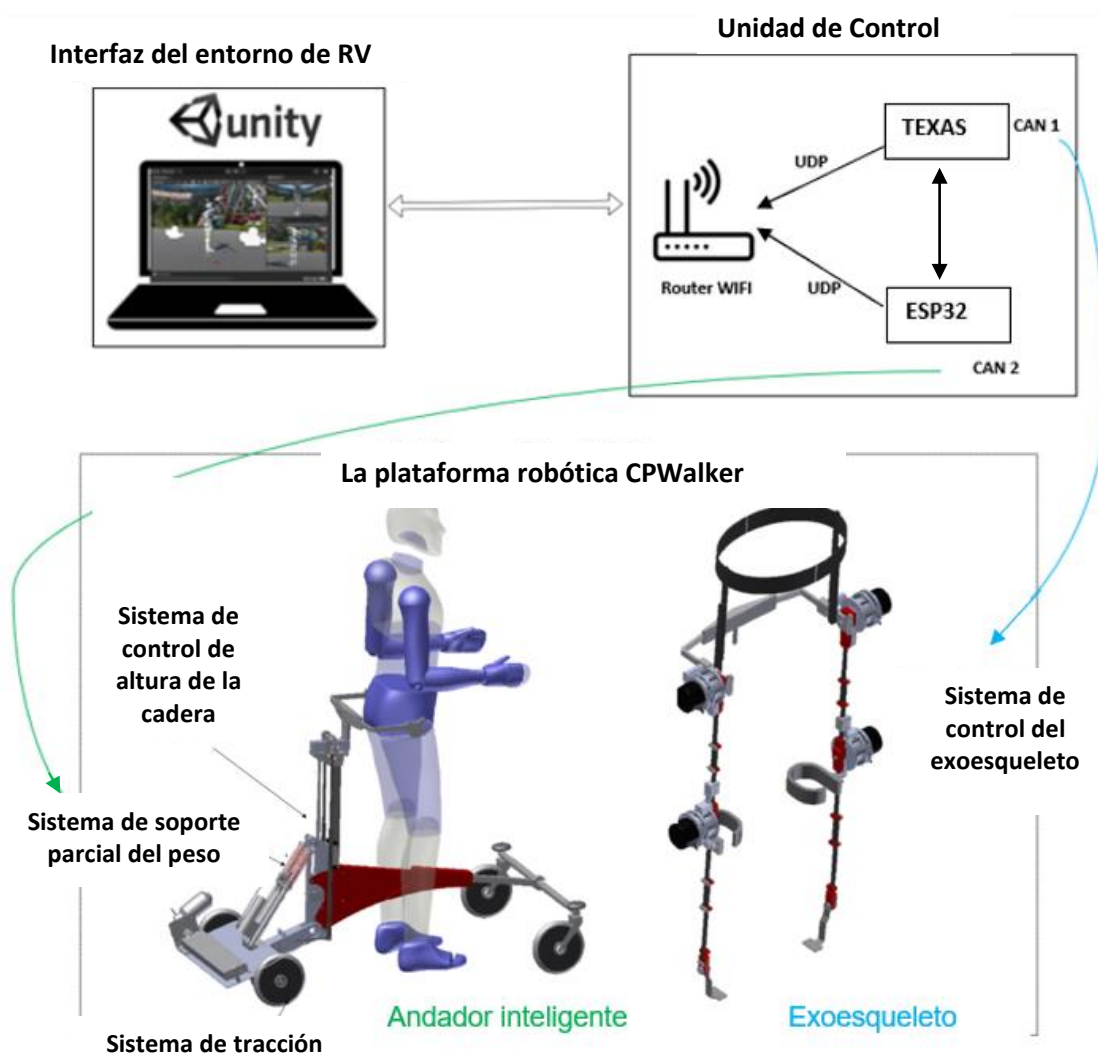


Figura 2.12. Esquema de la arquitectura de control general del CPWalker.



El protocolo de comunicación entre los sensores y la unidad de control es a través del bus CAN (Controller Area Network) y la comunicación se produce a una velocidad fija (1 ms) establecida por el esquema de la unidad de control [16].

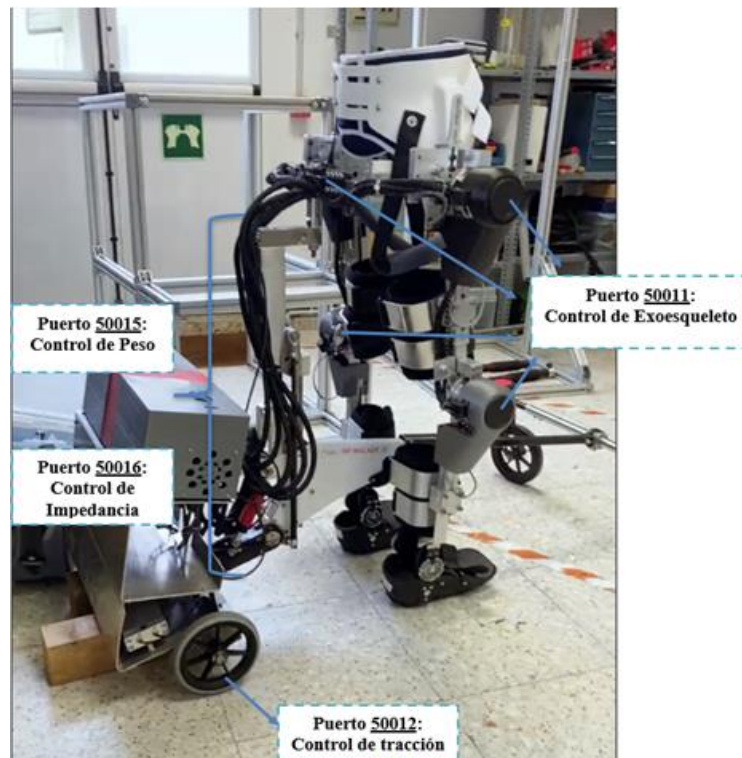
El ESP32 y la Texas se encargan de transferir las órdenes de control desde la interfaz del entorno virtual a los drivers de control, que son los que comandan tanto el exoesqueleto como el andador inteligente. La Texas recibe al mismo tiempo la información de los sensores de toda la plataforma a través del bus CAN.

En la [Figura 2.12](#), se puede apreciar que existen una Texas y un ESP32 que constituyen conjuntamente la unidad de control de la plataforma CPWalker. Asimismo, la Texas se comunicará con los sensores situados en ambas piernas del exoesqueleto a través del bus CAN 1.

Por otro lado, el andador inteligente está compuesto por el sistema de tracción, el sistema de soporte parcial del peso y el ajuste de la altura de las caderas. Los drivers para controlar los motores de estos sistemas y para leer sus sensores se comunicarán con la Texas a través del bus CAN2.

El sistema está conectado a una red Wifi que permitirá la comunicación del CPWalker con el ordenador donde se ejecuta el entorno virtual en Unity.

Se puede apreciar en la [Figura 2.12](#), la comunicación entre el entorno de realidad virtual y las unidades de control es través de red WLAN (Wireless Local Area Network). En total son cuatro puertos y cada uno de ellos se corresponde con un sistema activo del robot, como se ve en la [Figura 2.13](#).



*Figura 2.13. Los puertos asociados a cada sistema activo del CPWalker.*

El CPWalker recibe cuatro u ocho bytes en función del sistema activo y el puerto, el puerto de control de tracción recibe ocho bytes, mientras que el resto de los puertos reciben cuatro. A continuación, se muestra en la tabla (ver [Tabla 1](#)) los comandos correspondientes a cada puerto y sus posibles valores. Cada casilla de la tabla corresponde con un parámetro específico, las casillas marcadas con “xx” tendrán los bytes 0x00 por defecto y las casillas vacías indican que el rango de valores no es fijo.

Control del exoesqueleto						
<u>Puerto</u> <u>50011</u>	Articulaciones	Modo de control	Pasos	Cadencia	ROM (Rango del movimiento)	Longitud de las piernas (cm)
	LH RH LK RH 0 o 1	0, por defecto		[0-100]	[0-100]	
		4, control de posición				
		10, control de impedancia				
		20, posición inicial				
Control de tracción						
<u>Puerto</u> <u>50012</u>	Consigna de tracción		Manual Izquierda	Manual Derecha	Modo de tracción	
	≥ 100, posición inicial		[0,100] 50 = Stop		0, Stop	
	< 100, no posición inicial				10, Manual 20, Automático	
Control del peso del paciente						
<u>Puerto</u> <u>50015</u>	Calibrar	Peso del paciente (kg)		PBWS (%)	xx	
	0 o 1			[0-100]	0x00	
Control de impedancia						
<u>Puerto</u> <u>50016</u>	Calibrar	Nivel de impedancia		Comprobar galgas	Peso de referencia (kg)	
	0 o 1	0, 1, 2 y 3		0 o 1		

Tabla 1. Comandos de control de cada puerto del CPWalker.

- **El puerto de control de impedancia (puerto 50016)** permite ajustar el nivel de impedancia según tres niveles: alta impedancia (más próximo a un seguimiento de trayectoria puro), impedancia media e impedancia baja (más próxima al paciente como responsable del movimiento) [16]. La estrategia del control de impedancia tiene como objetivo incrementar la participación del paciente durante la terapia aprovechando su fuerza residual. Internamente se basa en un controlador de posición y fuerza en cascada, cuyo bucle es capaz de seguir los patrones de fuerza.
- **El puerto de control del peso (puerto 50015)** es responsable del control de descarga del peso del paciente (peso soportado por el CPWalker), a través del ajuste de unos parámetros (el peso del paciente y el porcentaje del peso de descarga) permite al usuario aprovechar su fuerza residual y mejorar la coordinación de los movimientos.
- **El puerto de control de tracción (puerto 50012)** controla las ruedas traseras del CPWalker y permite el movimiento de tracción. El primer byte se corresponde con la consigna de tracción, que permite reiniciar la posición (con un valor mayor o igual a 100); el segundo y el tercer byte controlan ambas ruedas traseras, estos van a compartir el mismo rango de valores de velocidades, entre 0 y 50 girarán en sentido antihorario, mientras que de 50 a 100 lo harán en sentido horario, siendo 50 el valor de parada. Cabe destacar que a cada rueda se le puede asignar un valor distinto, lo cual permite al CPWalker realizar giros mediante su tracción diferencial. El último byte se asocia con el modo de tracción, seleccionando entre los modos manual, automático o detención.
- **El puerto control del exoesqueleto (puerto 50011)** permite controlar el movimiento del conjunto del exoesqueleto teniendo en cuenta los ángulos de flexión y extensión de las caderas y las rodillas. Cada articulación del exoesqueleto tiene su propio controlador de posición e impedancia con parámetros específicos configurados individualmente para cada caso, por lo que se puede controlar cada parte del exoesqueleto por separado. Sin embargo, el seguimiento está garantizado en todo el exoesqueleto porque la referencia se envía para todos los controladores en cada ciclo [16]. Además, en el control del exoesqueleto se definen también el número de pasos, la cadencia de marcha, el rango de movimiento de las articulaciones, así como la longitud de las piernas del paciente.

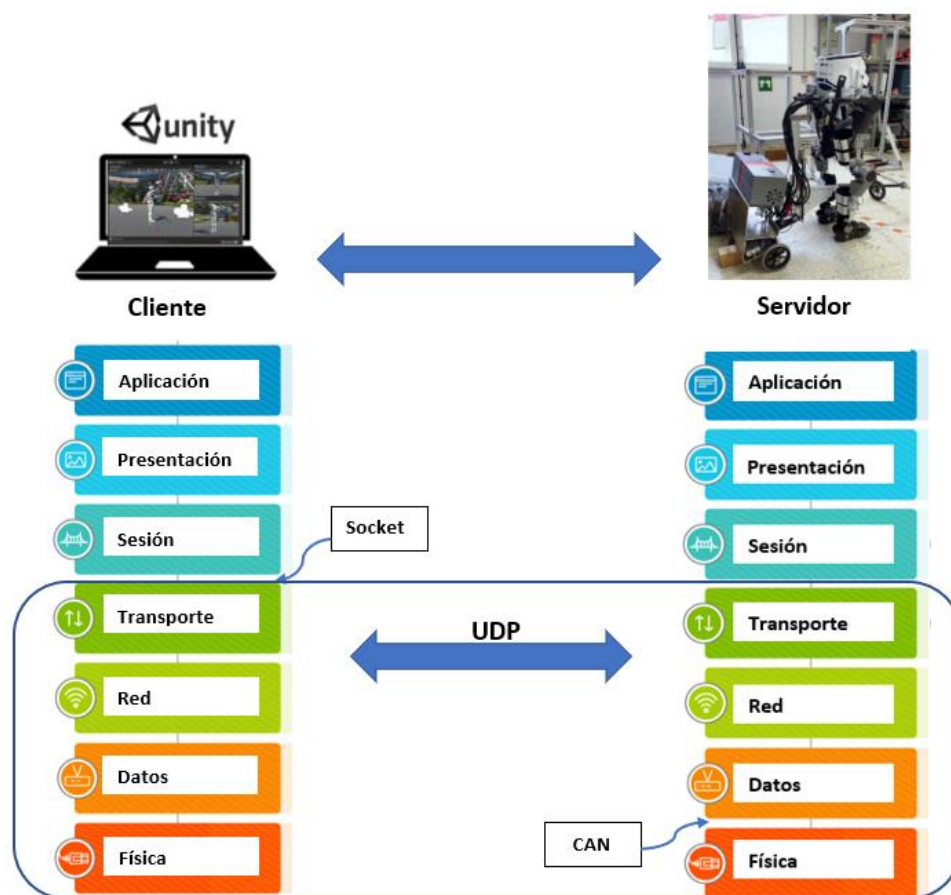


Figura 2.14. Capas de comunicación según el modelo OSI.

Como se ha comentado anteriormente, la comunicación entre Unity y el CPWALKER se establece a través de la red WiFi, mediante el uso de sockets de UDP (ver [Figura 2.14](#)). La ventaja de emplear el protocolo de UDP es que la velocidad y el flujo de información es rápida al prescindir de la verificación de ida y vuelta entre el dispositivo emisor y el receptor. Este aspecto es especialmente importante en la sincronización entre ambos sistemas, donde el envío de comandos de control y la recepción de datos tienen que ocurrir de forma casi simultánea para la actuación del CPWalker y la emulación de sus movimientos en el entorno virtual. Así pues, la velocidad de transmisión es más importante que la pérdida de algunos paquetes en tránsito.

Para ello, se usan las funciones de la clase *UDPClient*, pertenecientes al espacio de nombres de *System.Net.Sockets* [23]. La clase *UdpClient* proporciona métodos para enviar y recibir datagramas UDP a través de la red sin que se haya establecido previamente una conexión entre el emisor y el receptor. Asimismo, se ha definido una instancia de la *UdpClient* especificando un *AddressFamily* de tipo *InterNetwork*, para indicar que se espera una dirección IP versión 4 cuando el Socket se conecta a un punto de conexión; el *SocketType* de *Dgram* para indicar que admite datagramas y el *ProtocolType* *Udp* para indicar el protocolo de comunicación (ver [Figura 2.15](#)).



```
var ipCP = IPAddress.Parse("192.168.4.1");
const int port = 50012;
var udpClient1 = new Socket (AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);
var sendEndPoint = new IPEndPoint(ipCP, port);
udpClient1.SendTo( ConvertTo(valor), sendEndPoint);
```

*Figura 2.15. Fragmento de código que muestra la estructura del envío de órdenes de control.*

Los paquetes consisten en buffers de bytes que se envían mediante la función *SendTo()* a la dirección IP especificando el puerto de destino. Dentro de esta función se realiza una llamada a otra función *ConvertTo()* creada para la lectura de los valores, en decimal, de un comando en particular y su conversión a bytes, en hexadecimal, antes de ser enviado. Así, cada buffer de bytes forma un comando de control que actuará sobre el CPWalker según el puerto que le corresponda.

## 3.2 RECEPCIÓN Y DECODIFICACIÓN DE DATOS

Debido a ciertas limitaciones en el envío de datos desde el CPWalker, los datos se recibirán en sistema hexadecimal, pero para interpretarlos se ha de decodificar en el sistema decimal.

La [Figura 2.16](#) se muestra el esquema de conversión de los datos recibidos:

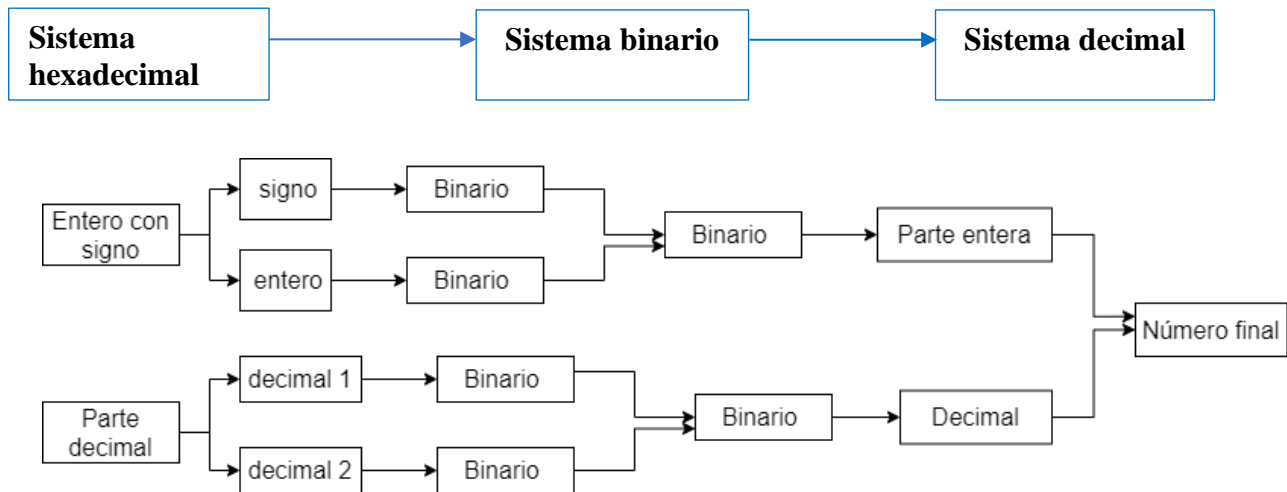


Figura 2.16. Esquema del sistema de conversión de datos.

1. Recepción y guardado del buffer de bytes hexadecimal.
2. Clasificación de datos según su tipo y se anota la posición que ocupa dentro del buffer de bytes.
3. Para los datos de tipo coma flotante, constan de dos partes, una parte entera con signo y otra decimal. Se ha de tener en cuenta el primer bit a la hora de pasar a binario ya que indica el signo.
4. Para los datos de tipo entero, no hace falta tener en cuenta el signo, ya que siempre serán positivos.

Para la recepción de datos del CPWalker se ha empleado el método *Receive()*. Este método se bloqueará hasta que llegue un datagrama desde el host remoto. Cuando los datos estén disponibles, *Receive()* leerá el primer datagrama en cola y devolverá la parte de datos como una matriz de bytes. Dado que estos bytes no siempre representan el mismo tipo de dato, la decodificación de los mensajes debe ser distinta uno respecto del otro.

Posición	Mensaje	Tipo de mensaje
1	Rodilla izquierda Real	Float
2	Rodilla izquierda Referencia	Float
3	Rodilla derecha Real	Float
4	Rodilla derecha Referencia	Float
5	Cadera izquierda Real	Float
6	Cadera izquierda Referencia	Float
7	Cadera derecha Real	Float
8	Cadera derecha Referencia	Float
9	Índice de la pierna derecha	Int
10	Índice de la pierna izquierda	Int
11	Torque de la rodilla izquierda	Float
12	Torque de la rodilla derecha	Float
13	Torque de la cadera izquierda	Float
14	Torque de la cadera derecha	Float
15	Peso	Float
16	Pistón	Int
17	Velocidad de tracción	Float
18	Encoder de la rueda izquierda	Float
19	Encoder de la rueda derecha	Float

Tabla 2. de referencia del tipo de mensaje que representa cada byte y su indexación.

Como se puede apreciar en la tabla de la [Tabla 2](#), cada posición de la matriz de bytes se corresponde con un dato diferente, y cada uno puede ser de tipo coma flotante o entero. Así, los mensajes de tipo coma flotante requieren 4 bytes para su codificación mientras que los de tipo entero ocupan 1 byte. En total suman 67 bytes y se ha comprobado con *Wireshark* los datagramas de UDP que están llegando al PC donde se desarrolla el entorno virtual (ver [Figura 2.17](#)).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.100	192.168.1.101	UDP	109	62512 → 10000 Len=67
2	0.010028	192.168.1.100	192.168.1.101	UDP	109	62512 → 10000 Len=67
3	0.017414	192.168.1.100	192.168.1.101	UDP	109	62512 → 10000 Len=67
4	0.027196	192.168.1.100	192.168.1.101	UDP	109	62512 → 10000 Len=67
5	0.037429	192.168.1.100	192.168.1.101	UDP	109	62512 → 10000 Len=67
6	0.052678	192.168.1.100	192.168.1.101	UDP	109	62512 → 10000 Len=67

>	Frame 1: 109 bytes on wire (872 bits), 109 bytes captured (872 bits) on interface \Device\NPF_{0CFEAAFA}
>	Ethernet II, Src: Espressi_00:13:fc (fc:f5:c4:00:13:fc), Dst: IntelCor_c0:0d:0f (0c:7a:15:c0:0d:0f)
>	Internet Protocol Version 4, Src: 192.168.1.100, Dst: 192.168.1.101
>	User Datagram Protocol, Src Port: 62512, Dst Port: 10000
>	Data (67 bytes)
	Data: 0900fa0200000000008001b020000000f8ff73020000000fcff6402000000000000100...
	[Length: 67]

Figura 2.17. Datos de UDP capturados con Wireshark.

Para la decodificación de los datos recibidos se escribió la función *DecodifiedMessage()* a la que se le pasa como parámetro la matriz de bytes obtenida con la función *Receive()*, el pase de parámetros es por valor, a no ser que se indique lo contrario. De esta manera, C# permite trabajar directamente con dichos bytes.

Cabe destacar que dentro de la función *DecodifiedMessage()* se hacen llamadas a otras dos funciones, *Data2Double()* y *Data2Integer()*, creadas para la decodificación de datos según su tipo. Cada byte equivale a 8 bits, y un dato de tipo “float” necesita 4 bytes para su decodificación, razón por la que se recorre los primeros 32 bits con un contador que incrementa de 4 en 4, que es hasta donde llegan los datos de tipo “float” (ver [Figura 2.18](#)).

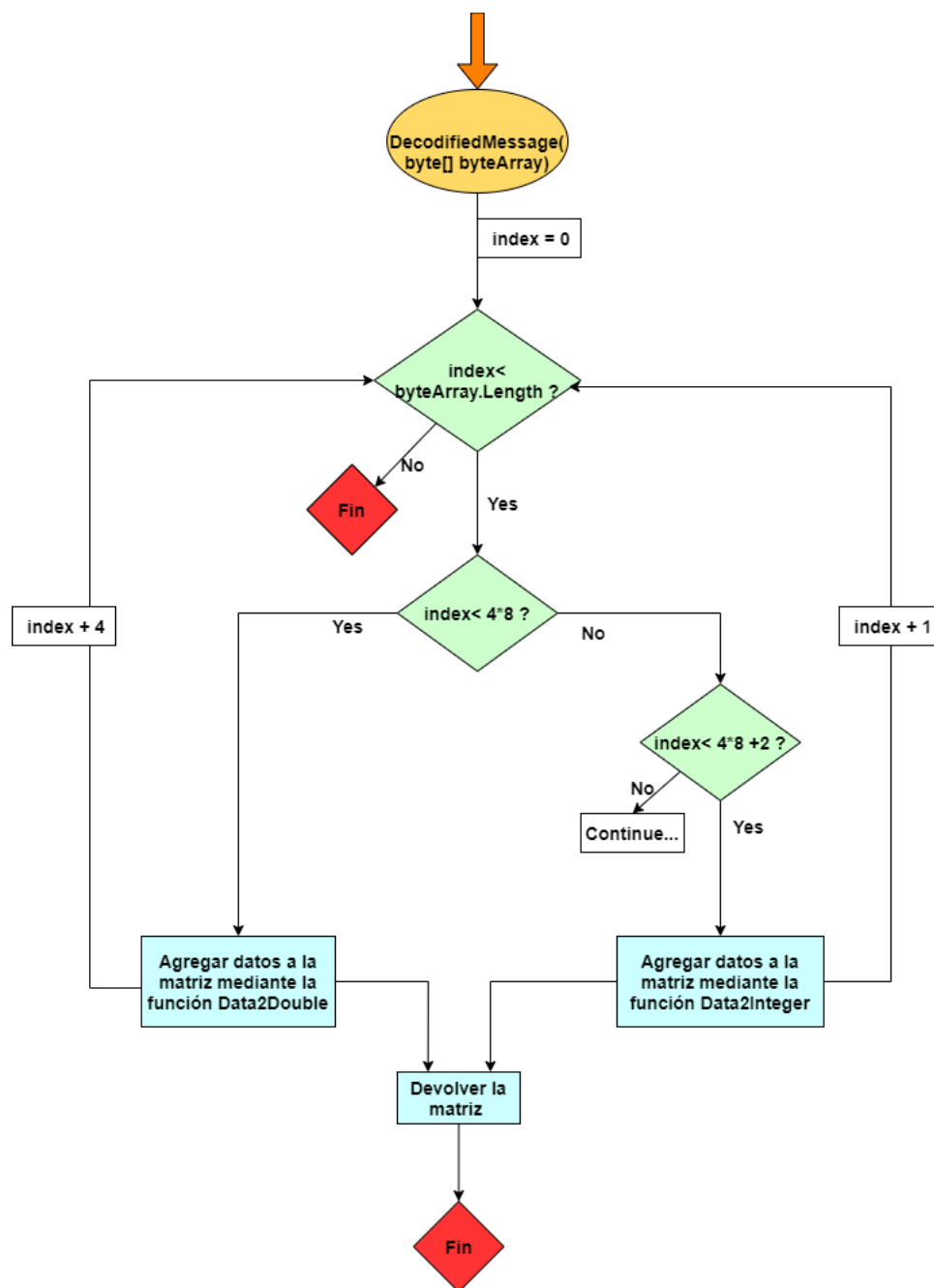


Figura 2.18. Diagrama del código que se ejecuta dentro de la función *DecodifiedMessage()*.

Por otra parte, la función que se ha empleado para la decodificación de datos “float” consiste en pasar los bytes a binario y agruparlos según si representan la parte entera o la parte decimal del número ( ver [Tabla 3](#)).

Mensaje de tipo coma flotante			
Integer (int16)		Decimals (int 16)	
Integer1 (int 8)	Integer2 (int 8)	Decimal1 (int 8)	Decimal2 (int 8)
[0 -255]	[0 -255]	[0 -255]	[0 -255]
Signo	Entero	Decimal	Decimal

Mensaje de tipo entero
Integer (int 8)
[0-255]

Tabla 3. La decodificación de los datos según su tipo

La ventaja de programar en C# es su capacidad de tratar cualquier tipo de dato simple como objeto, lo cual permite emplear métodos asociados a un tipo de dato “int” y aplicar el método *ToString()* para obtener un valor “string” de ese “int”, así como concatenar dos cadenas de caracteres. Así pues, se ha aplicado este concepto para la transformación de datos de la siguiente manera:

En primer lugar, se han creado variables para convertir cada byte a una cadena de binarios a través de la función *ToString()* indicando la base a la que se quiere transformar, luego se concatenan los 2 bytes del mismo tipo y finalmente se aplica *ToInt16()* para obtener el valor final. Para obtener la parte decimal se ha tenido que dividir el valor final entre mil para obtener las cifras decimales correspondientes .

Para los números positivos se suman la parte entera y decimal correspondiente, mientras que para los negativos se ha de tener en cuenta la primera cifra, ya que se trata del bit de signo (el bit menos significativo). Un número es negativo si y sólo si su bit menos significativo es igual a “1” . Teniendo esto en cuenta, se ha escrito el código correspondiente para realizar la operación en complemento a dos (ver [Figura 2.19](#)).

Por último, los datos decodificados se almacenan en una matriz, siguiendo la misma indexación que los recibidos del CPWalker, y serán utilizados para simular el movimiento de las piernas en el entorno virtual.

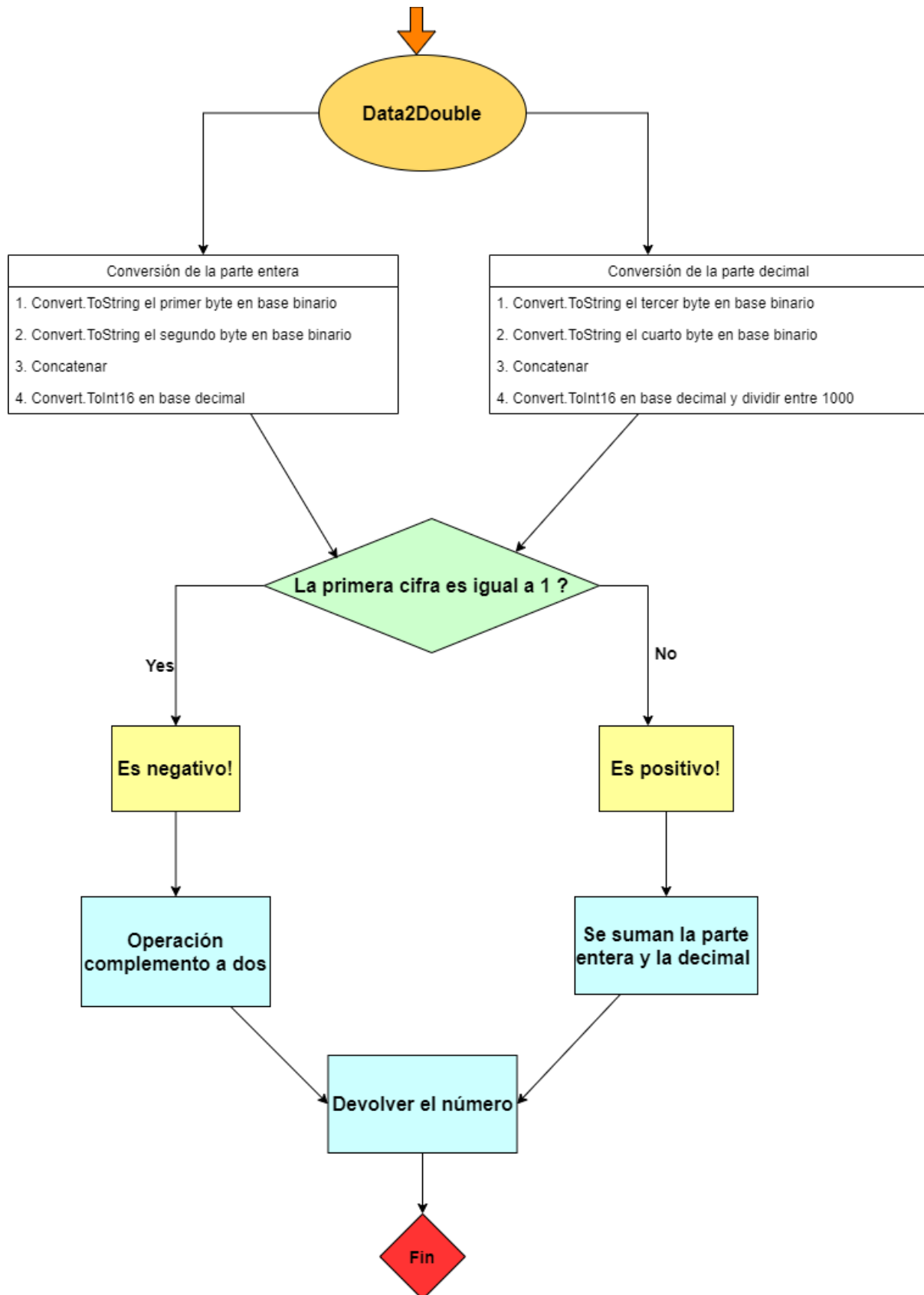


Figura 2.19. Diagrama del código para tratar con los números negativos.

### 3.3 LA CINEMÁTICA EN UNITY

Una vez decodificados los datos del CPWALKER y definidas las articulaciones que van a ser procesadas (las de la cadera y rodillas), se ha de extraer la información de la matriz de datos decodificados que se guardó anteriormente.

El movimiento del avatar en Unity es producido por la cinemática directa, la cual permite conocer las posiciones de los efectores finales a partir de los ángulos conocidos de las articulaciones. Para ello, se acceden a las posiciones correspondientes de la matriz de datos decodificada para obtener los ángulos correspondientes.

El objetivo es emplear los ángulos generados por el CPWalker junto con el paciente en tiempo real. Debido a que no se ha podido simular con pacientes, en este trabajo se han empleado unos ángulos definidos por un patrón de marcha predeterminado. Esta estrategia corresponde con las primeras fases de rehabilitación, donde el objetivo es guiar a los miembros inferiores del paciente siguiendo los patrones de marcha. Estos ángulos corresponden con los índices 1, 3, 5, 7 de la matriz de datos decodificada.

A la hora de aplicar estos datos en el avatar, se tuvo en cuenta que las articulaciones de la cadera son de tipo esférica. Tienen tres grados de libertad y permite la flexión/extensión, la abducción/aducción y la rotación externa/interna. Las articulaciones de las rodillas son de tipo bisagra, tienen un grado de libertad, y permiten movimientos de flexión y extensión, los cuales se asocian con las rotaciones según el eje X del entorno virtual (ver [Figura 2.20](#)).

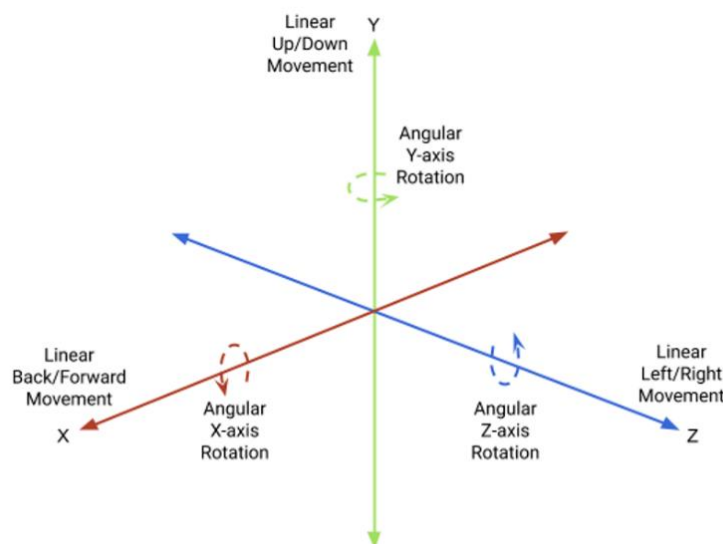


Figura 2.20. El sistema de referencia en Unity3D [22].

Para simular la cinemática se ha tenido que aplicar propiedades a los objetos. Las propiedades más interesantes son los *joints*, entendido como uniones entre los objetos y permiten restringir el movimiento de algunos cuerpos rígidos con respecto a otros. Estos

pueden ser: *Fixed joint*, *Hinge joint*, *Spring joint*, *Character joint* y *Configurable joint*, cada una de ellas posee unas características específicas, siendo la última de la lista la más versátil.

Atendiendo a las articulaciones reales que se pretenden emular, se han seleccionado articulaciones de tipo *Character joint* que simulan articulaciones esféricas (ball and socket joint), la cual simularía a las articulaciones de la cadera; y el *Hinge joint* o articulación de bisagra que simularía la rodilla. Sin embargo, este planteamiento fue descartado tras un exhaustivo estudio y simulación con los datos del CPWalker. Puesto que al aplicarlos en conjunto sobre el avatar producía movimientos irreales (ver [Figura 2.21](#)). Según la documentación de Unity, cada “*joint*” se descompondrá en dos restricciones primitivas, una restricción lineal para mantener los cuerpos a cierta distancia específica y una restricción angular para mantener los cuerpos orientados de una manera específica alrededor de un eje particular. Estos movimientos son impuestos por el motor de física, que solo garantiza la satisfacción de restricciones si encuentra una solución que es capaz de converger tras un conjunto de iteraciones [25]. En este caso, al imponer distintos tipos de articulaciones dentro del mismo “*root*”, el motor de física deja ciertas restricciones fuera provocando un movimiento de giro fuera del rango impuesto.



*Figura 2.21. Movimientos no realistas con los joints regulares de Unity.*

En su lugar, se emplean los cuaterniones para simular un comportamiento físico realista. La ventaja de emplearlos es que tienen una menor carga computacional a la hora de ejecutar los movimientos y previene el error debido al bloqueo de cardán (“Gimbal lock”) que se produce cuando al rotar los ejes el sistema pierde la referencia de al menos uno de ellos, haciendo que cuando rote sobre un determinado eje, se modifique de manera paralela los datos de los otros dos ejes simultáneamente. El uso de cuaterniones evita este tipo de error al tener una cuarta componente.



A pesar de que Unity almacena todas las rotaciones de *GameObject* internamente como cuaterniones, el inspector muestra la rotación utilizando ángulos de Euler, porque es más intuitivo y fácil de editar. Para ello, Unity hace operaciones internas para dicha conversión [26].

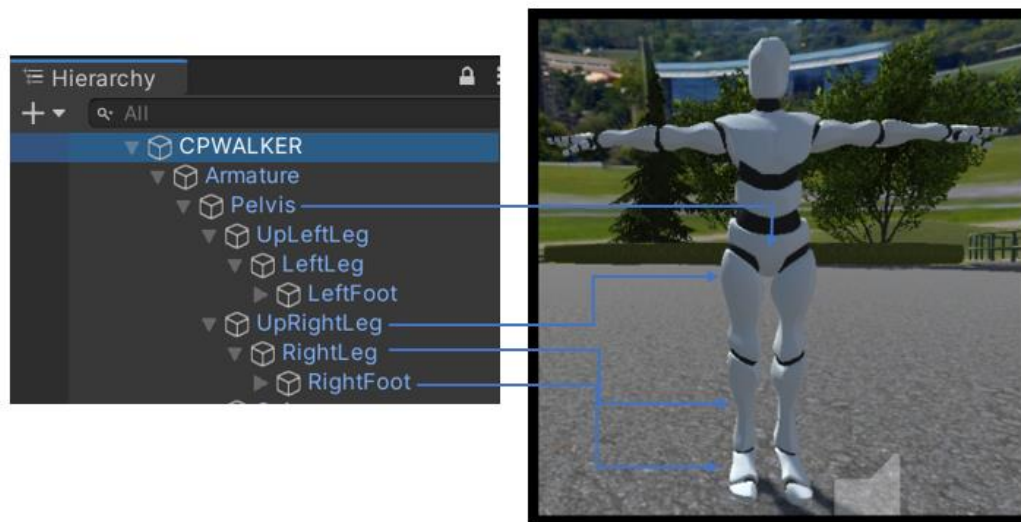


Figura 2.22. Jerarquía de las articulaciones del avatar.

Para aplicar los cuaterniones se debe acceder al *Transform* de los *GameObjects*, es decir, a las articulaciones del avatar y en particular, a sus orientaciones. Sin embargo, debido a la jerarquía del modelo, no es posible acceder directamente a un nodo inferior sin acceder primero a un nodo superior (ver la Figura 2.22). Para solventar esto se empleó la función *GetComponentInChildren()* para buscar los nodos hijos de la raíz (pelvis) a través de su nombre de referencia (ver Figura 2.23).

Una vez hecho esto, ya se puede acceder a cada nodo individualmente para aplicar los cuaterniones en forma de *Quaternion.Euler()* a las propiedades de *Transform.rotation* y *Transform.localRotation* de cada articulación, mediante los datos decodificados del CPWalker.

Los ángulos de las articulaciones en Unity están definidos en grados, al igual que los datos procedentes del CPWalker, por lo que se puede aplicar directamente sin realizar ninguna conversión adicional. En este punto es importante tener en cuenta la propiedad *axis* presente en cada articulación que se explicó anteriormente (ver Figura 2.20), ya que proporciona información sobre el eje en el que se debe aplicar la rotación, en este caso, es el eje X.

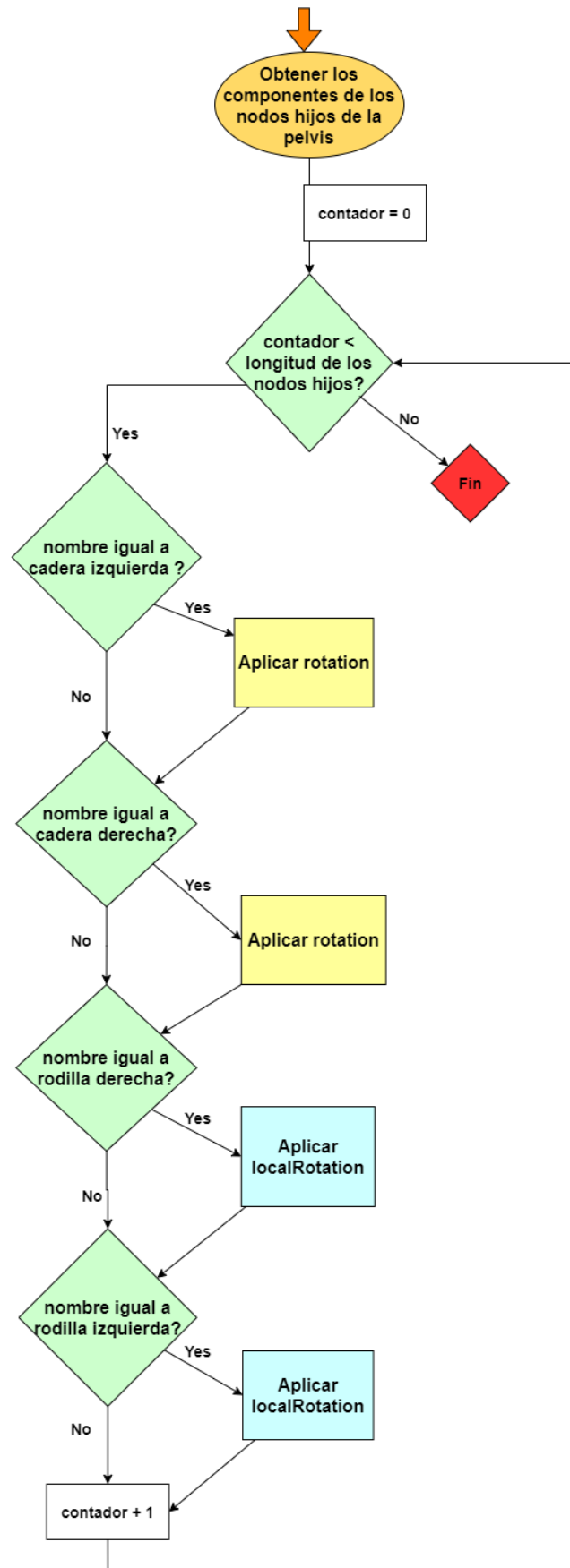
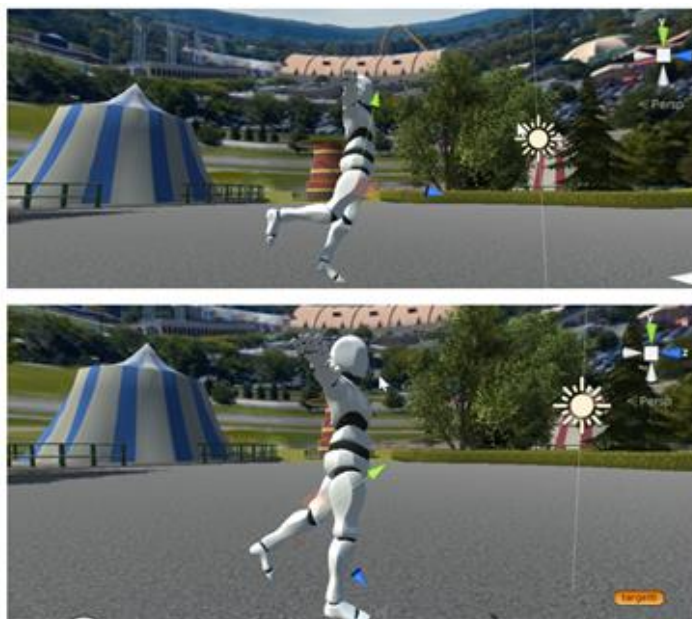


Figura 2.23. Diagrama del código para encontrar los nodos hijos y aplicar las orientaciones correspondientes.

Otro punto por mencionar es que, debido a la jerarquía (ver la [Figura 2.22](#)), la posición de una articulación cambia de acuerdo con la rotación de su padre. Razón por la que solo se ha aplicado *localRotation* en las rodillas, ya que además de mover con respecto a las caderas poseen sus propias rotaciones. Análogamente, la posición y orientación del efector final (manos, pies) de una cadena de articulaciones puede ser determinado a partir de los ángulos y las posiciones relativas de los nodos del nivel superior. Este es el caso de los pies, a los que no se han aplicado ninguna rotación, pero son capaces de moverse conforme a las orientaciones de las piernas.

La [Figura 2.24](#) muestra algunas capturas de los movimientos del avatar a través de los datos recibidos del CPWalker.



*Figura 2.24. Movimiento de las piernas del avatar capturados en distintos fotogramas.*

Tras las simulaciones realizadas se puede comprobar que, mediante una aplicación correcta de las propiedades físicas, el sistema es capaz de emular los movimientos del CPWalker con los datos decodificados.

## 4. CONCLUSIONES

A lo largo de este capítulo se han abordado los aspectos más importantes del desarrollo de este TFG para lograr la conexión del CPWalker con el entorno virtual. Desde el diseño del mundo virtual que permitirá el control del CPWalker mediante el envío de secuencias de órdenes, hasta el tratamiento de los datos de realimentación del CPWalker con el fin de aplicar las orientaciones adecuadas en el avatar y lograr así la sincronización del movimiento de ambos sistemas.

## IV. RESULTADOS

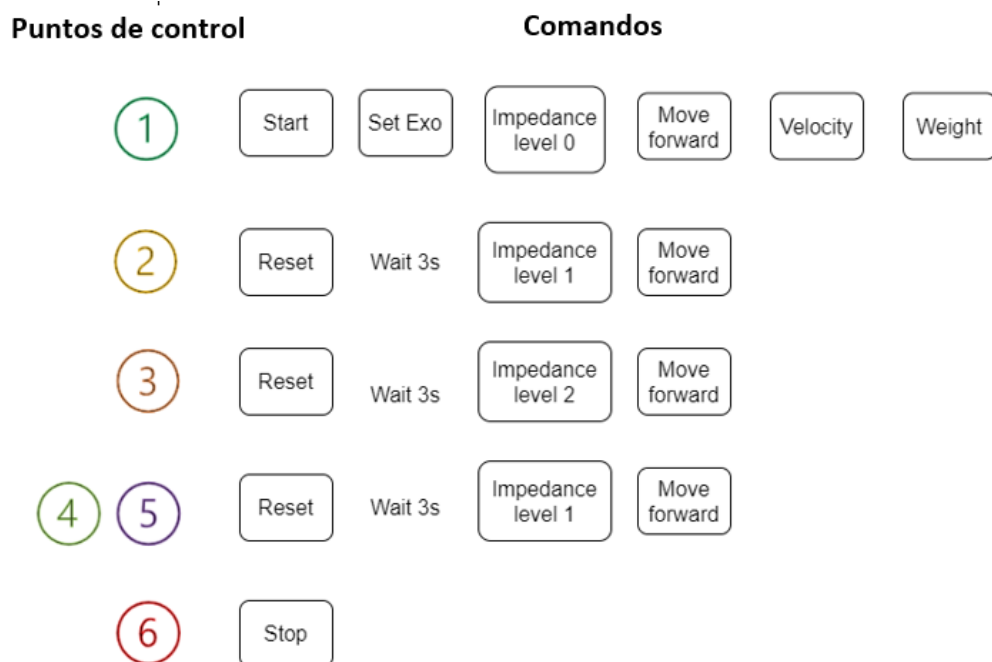
Una vez estudiado y comprobado que los distintos módulos funcionan correctamente por separado, se ha procedido a ensamblarlos en el proyecto final.

### 1. DEFINICIÓN DE LA SIMULACIÓN

Se ha generado una trayectoria con varios puntos de control, de manera que el avatar debe seguir el camino definido por esos puntos. Paralelamente, en cada punto de control se han activado mecanismos de detección de colisión con el avatar. De manera que cuando el avatar atraviesa un determinado punto de control, se enviarán unas órdenes al CPWalker con parámetros predefinidos. Por lo que, si se emplease en una sesión de rehabilitación, el terapeuta no tendrá que realizar más acciones que dar el botón de inicio.

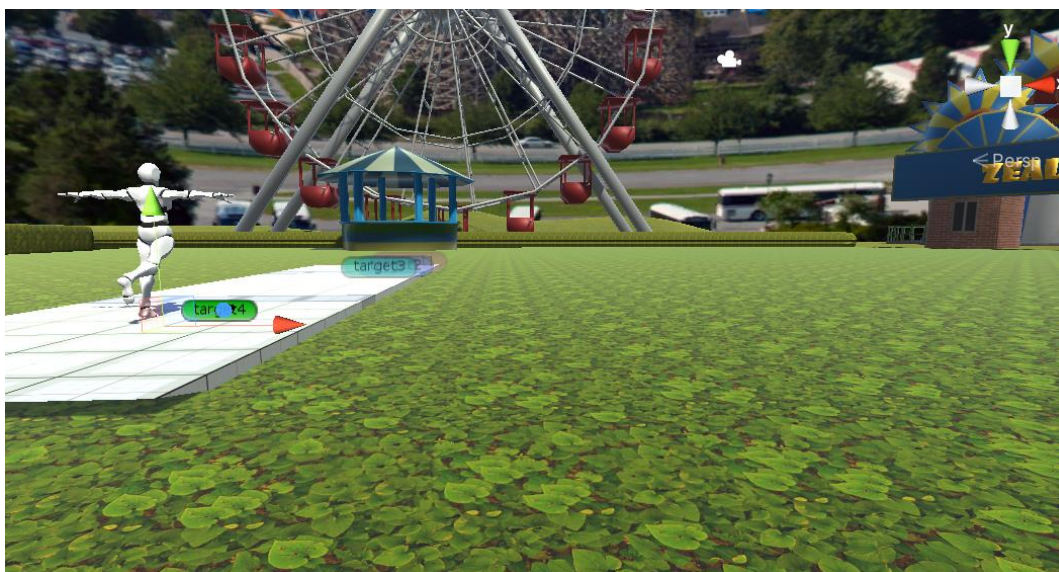
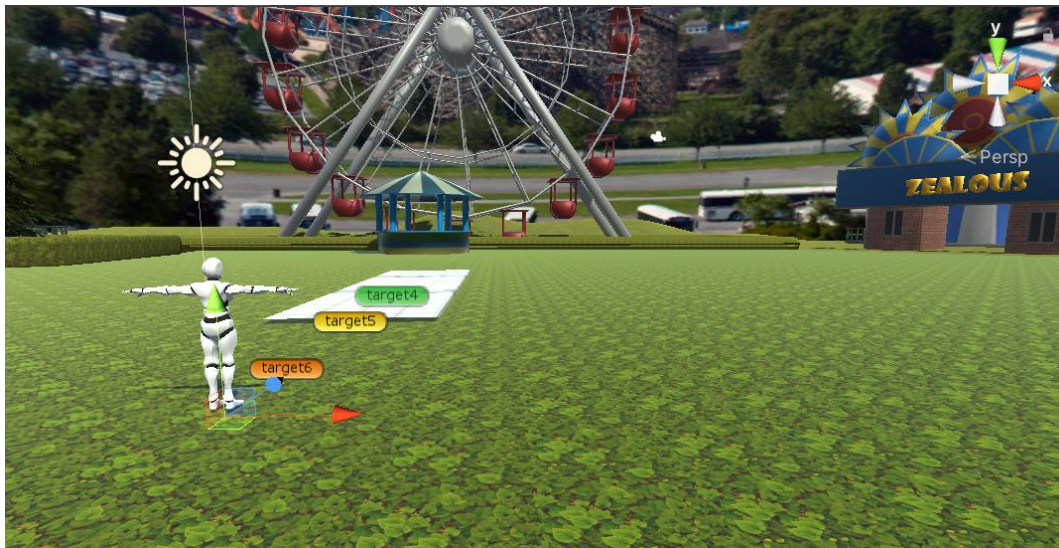
En caso de que se desee detener la sesión, se podrá hacer de dos maneras. Por un lado, con el botón de parada de emergencia situada en la interfaz del entorno virtual, que envía el comando de parada al CPWalker, pudiendo retomar la acción en donde se había dejado. Por otro lado, se puede parar directamente la ejecución de Unity, pero al reiniciar la simulación se debe empezar la sesión desde el principio.

En las *Figuras 3.1* y *3.2* se muestran los comandos empleados para que el avatar siga la trayectoria definida por los puntos de control y emulando los movimientos del CPWalker. En total hay seis puntos de control y tanto estos como los comandos pueden ser modificados para adaptarse a las necesidades de cada paciente.



*Figura 3.1. Ejemplo de comandos empleados en cada punto de control.*







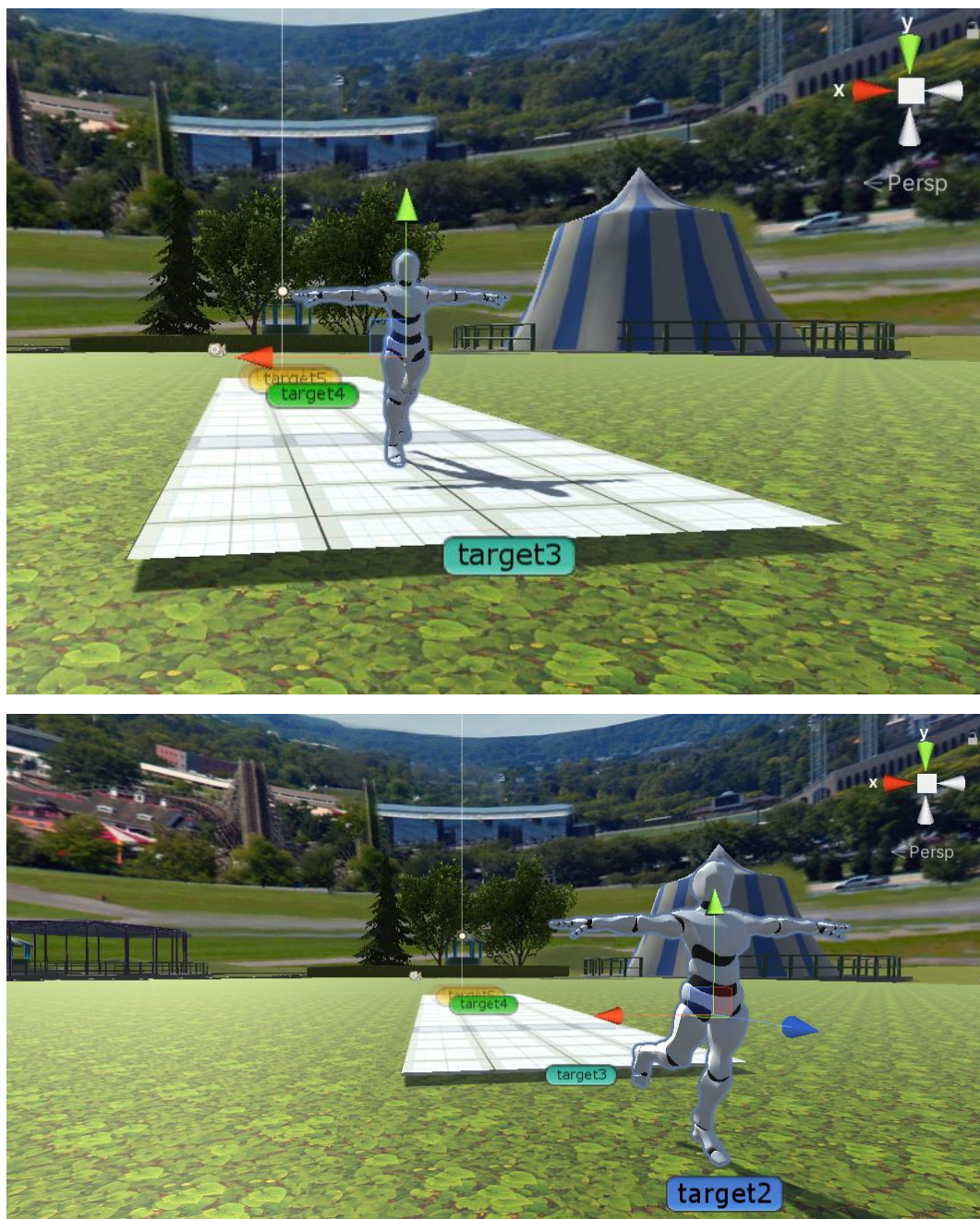


Figura 3.2 El avatar en distintos instantes del tiempo de la simulación.

## 2. ADQUISICIÓN Y ANÁLISIS DE LOS DATOS

Unity permite extraer información de la ejecución y pasarla a un archivo a través de la biblioteca *System.IO*. Asimismo, dentro de la función *FixedUpdate()*, se ha escrito el código correspondiente para acceder a los ficheros y escribir los datos en ellos.

Esta función se actualiza cada 0,02s por defecto. Para reducir la cantidad de datos iguales que se guardan durante la ejecución, se ha creado una variable que actúa como contador de *Time.deltaTime\*5* (ver [Figura 3.3](#)). En lugar de imprimir 50 veces, se imprimen 10 veces los datos, como se puede observar en la [Figura 3.4](#). Cada fichero está formado por cinco columnas, la primera muestra el instante de tiempo, y las otras cuatro representan los ángulos de las articulaciones caderas y rodillas izquierda y derecha.

```
var t = Time.deltaTime*5;
tiempo += t;
using (StreamWriter writer = File.AppendText(saveFilePath))
{
    string line = string.Format("{0:F2} {1:F3} {2:F3} {3:F3}
{4:F3}",
        tiempo,
        jointList[0].transform.localEulerAngles.x,
        jointList[1].transform.localEulerAngles.x,
        jointList[2].transform.localEulerAngles.x,
        jointList[3].transform.localEulerAngles.x
    );
    writer.WriteLine(line);
}
```

*Figura 3.3 Fragmento de código para generar el fichero correspondiente.*

Una vez finalizada la terapia se puede acceder a los archivos que contienen la orientación de las articulaciones de las caderas y las rodillas en cada instante de tiempo, tanto de los datos recibidos del CPWalker como de los ángulos en los que se mueven en el entorno virtual. La estructura del fichero se muestra en la [Figura 3.4](#):

Tiempo	Cadera izquierda	Cadera derecha	Rodilla izquierda	Rodilla derecha
15,80	34,233	12,175	48,251	9,179
15,90	34,233	12,175	48,251	9,179
16,00	34,233	12,175	48,251	9,179
16,10	38,112	7,151	42,165	10,168
16,20	38,112	7,151	42,165	10,168
16,30	38,112	7,151	42,165	10,168
16,40	38,112	7,151	42,165	10,168
16,50	38,112	7,151	42,165	10,168
16,60	38,112	7,151	42,165	10,168
16,70	38,112	7,151	42,165	10,168
16,80	38,112	7,151	42,165	10,168
16,90	38,112	7,151	42,165	10,168
17,00	38,112	7,151	42,165	10,168
17,10	38,112	7,151	42,165	10,168

Figura 3.4. Fichero con ángulos en los que se mueven las articulaciones del CPWalker.

Respecto a los datos obtenidos, cabe destacar que la orientación de los ángulos de las articulaciones del CPWalker es únicamente en el plano sagital (ver [Figura 3.5](#)).

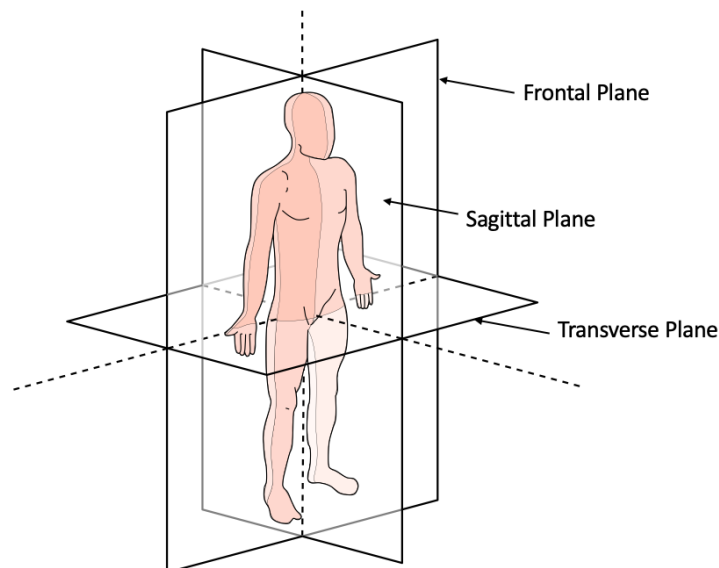


Figura 3.5. Los planos anatómicos del cuerpo humano [27].

Por ello, aunque las articulaciones de la cadera tienen tres grados de libertad, solo se van a mover en la dirección del eje X, como se puede comprobar en los ángulos obtenidos a través de Unity (ver [Figura 3.4](#)), ya que solo presentan el componente según el eje X.

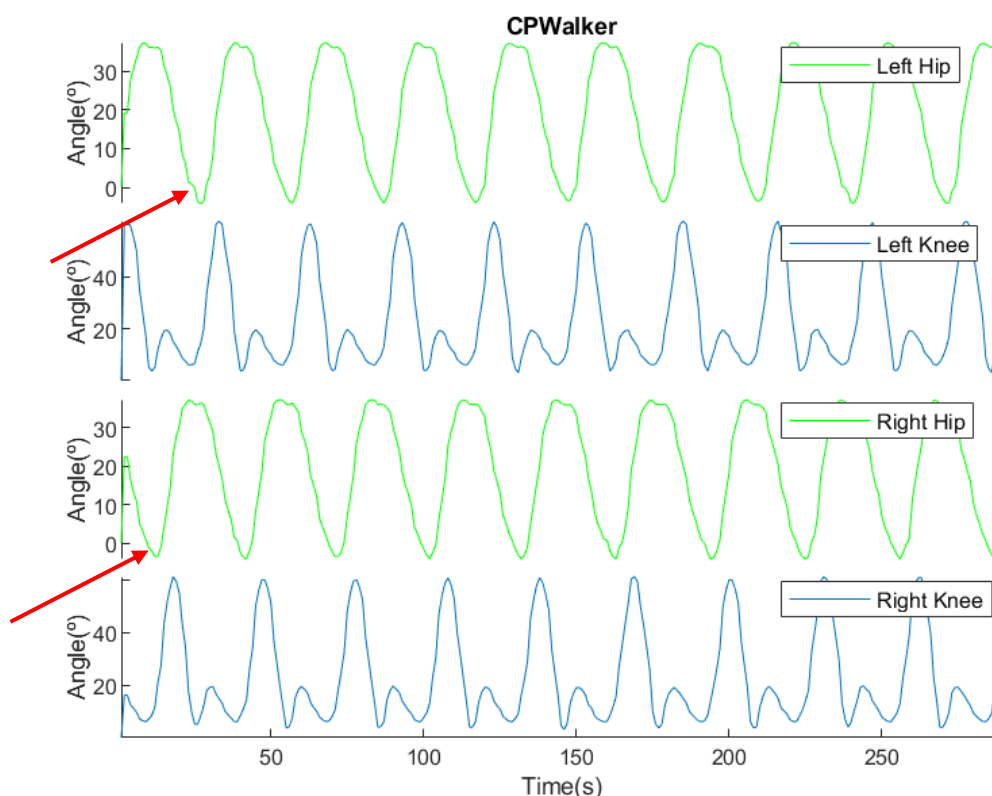
Por otra parte, Unity interpreta los ángulos negativos respecto al eje transversal del cuerpo como  $360^\circ - \text{ángulo}$ , por ejemplo, si el ángulo que recibe del CPWalker son  $-5^\circ$ , lo toma



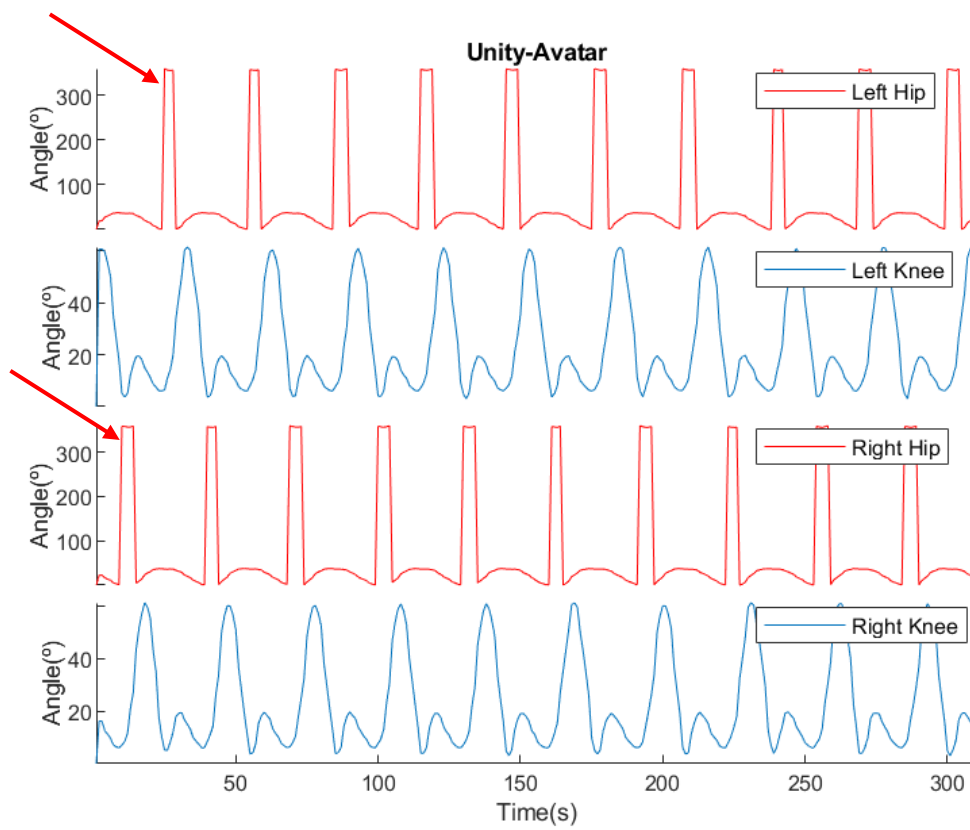
como  $355^\circ$ . En las gráficas de las *Figuras 3.6* y *3.7* se puede apreciar que las caderas del avatar presentan unos picos (señalados con flechas rojas) ya que los ángulos son negativos en el CPWalker.

La obtención y guardado los ángulos de flexión-extensión de las diferentes articulaciones sirven para contrastar con los datos reales. Así, nos asegura que las orientaciones del avatar son correctas, o al menos, no se perciben datos falsos.

Como apunte final, remarcar que, al operar solo con los miembros inferiores del avatar, el cuerpo superior queda inmóvil, adquiriendo esa posición rígida que tiene el avatar al inicio de la configuración y no permite simular el balanceo natural de los brazos durante la marcha. Esto se debe a que no se disponen los datos necesarios para la simulación del balanceo de los brazos. A pesar de ello, se han conseguido resultados satisfactorios, siendo un sistema de bajo coste, con características personalizables y basada en la realidad virtual. Lo cual sirve como herramienta de apoyo a la rehabilitación basada en la robótica, aumentando la motivación de los pacientes en su proceso de rehabilitación.



*Figura 3.6. Ángulos de flexión-extensión de las articulaciones del CPWalker*



*Figura 3.7. Ángulos de flexión-extensión del avatar en el entorno virtual.*

## V. CONCLUSIONES Y LÍNEAS FUTURAS

### 1. CONCLUSIONES

Este trabajo se ha enfocado en el diseño e implementación de tecnologías de realidad virtual junto con la robótica como parte de la rehabilitación de niños con parálisis cerebral. Se ha desarrollado una simulación de la sesión de terapia conectando la plataforma robótica CPWalker y el entorno de realidad virtual.

La interfaz del entorno virtual se ha desarrollado en Unity3D para así mostrar el modelo del avatar y poder simular la sincronización del movimiento de las extremidades inferiores del exoesqueleto del CPWalker. El entorno virtual conectado a la red WLAN del CPWalker recibe datos de las posiciones actuales del robot en tiempo real y envía órdenes de control al CPWalker (el soporte del peso, el nivel de impedancia, la velocidad de tracción, etc) a través del protocolo de comunicación UDP. Tanto para la lectura como el envío de datos se requiere una fase previa de conversión de estos. Finalmente, se toma la realimentación de los datos de orientación de las articulaciones del CPWalker para aplicarlo en el avatar, provocando un desplazamiento en el motor gráfico. Por tanto, se puede concluir que se ha alcanzado los objetivos propuestos de manera exitosa.

Una de las tareas inmediatas de este proyecto es experimentar con los pacientes durante una sesión de rehabilitación estándar y probar el rendimiento del sistema. Con la realimentación tanto de los pacientes como de los profesionales sanitarios se podrán realizar cambios oportunos para adaptar a las necesidades clínicas y mejorar el cuidado del paciente.

#### 1.1 OTROS ASPECTOS INTERESANTES DEL DESARROLLO

En esta sección se recogen los aspectos del desarrollo del proyecto que me han parecido interesantes, pues todo ello constituye una parte esencial en la realización de este trabajo.

En primer lugar, para el correcto funcionamiento de las distintas componentes en conjunto, se ha realizado y comprobado la actividad de cada una de ellas de manera individual, mediante modo *Debug* con el fin de detectar posibles errores de programación y corregirlos antes de su integración con el resto. A pesar de ello, algunas mejoras fueron introducidas después de realizar el estudio global. Este es el caso de la decodificación de los datos, inicialmente se llevó a cabo fuera del entorno virtual, en un IDE o entorno de desarrollo integrado (JetBrains Rider) puesto que no se necesitan funciones propias de Unity.

A continuación, para la sincronización del movimiento del avatar con el CPWalker, se ha montado otra escena sólo con componentes básicos (ver [Figura 4.1](#)) ya que el proyecto principal está compuesto por numerosos objetos, lo cual hace que su ejecución conlleve un mayor coste computacional. Asimismo, antes de realizar pruebas con el CPWalker, se han

hecho simulaciones con un modelo de Matlab-Simulink, el cual envía los datos de UDP exactamente igual que el CPWalker. Tras comprobar que la simulación se ha llevado a cabo de manera satisfactoria, el siguiente paso fue definir la simulación final con todos los componentes integrados, así como la adquisición y análisis de datos obtenidos, que se explicó en el capítulo de resultados.

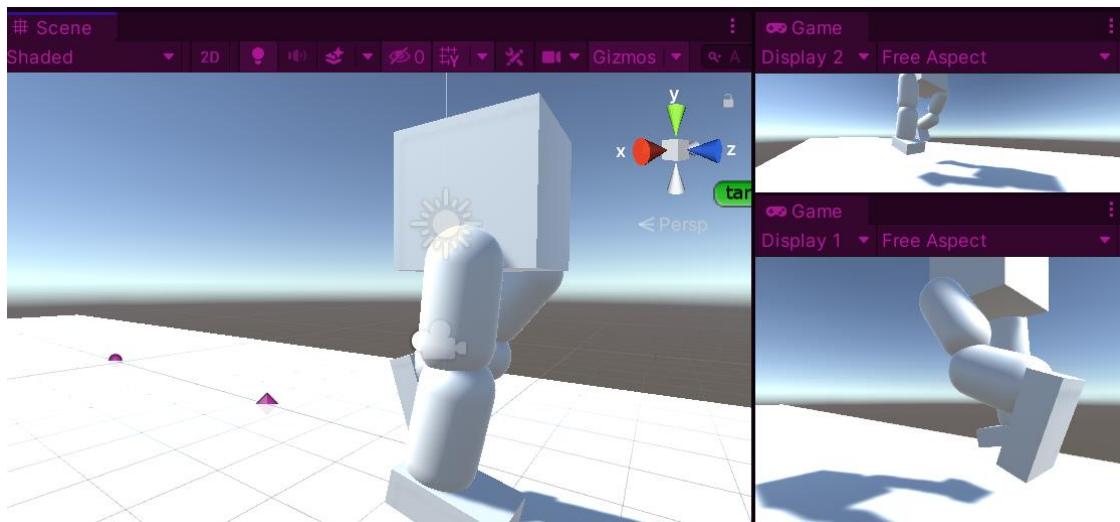


Figura 4.1. El modelo simplificado para la sincronización del movimiento de los miembros inferiores.

Durante el desarrollo del proyecto se ha considerado que puede ser útil la implementación de una interfaz de interacción con el usuario dentro del entorno virtual. Ésta permite actuar sobre el control de tracción, modificando la dirección y la velocidad del CPWalker de manera rápida y sencilla (ver [Figura 4.2](#)). Consiste en cuatro botones con sus etiquetas correspondientes según la dirección de movimiento y una barra deslizante para ajustar la velocidad de tracción, la cual comienza con 0 por defecto. Adicionalmente, se han creado otros dos botones, uno de reinicio, necesario para volver a la posición inicial y otro de parada de emergencia para la seguridad del paciente. Esta interfaz no fue incluida en la simulación final ya que se consideró que podría interferir en la ejecución del resto de comandos.



Figura 4.2. Interfaz de control de tracción renderizada con una cámara de la escena.

## 2. LÍNEAS FUTURAS

A pesar de alcanzar los objetivos propuestos, se considera que este sistema puede mejorar en muchos aspectos. Desde una mejora técnica del propio entorno virtual, como la inclusión de nuevas funcionalidades, hasta mejoras en la sincronización completa con el CPWalker.

En este trabajo se ha conseguido utilizar la realimentación de los ángulos medidos desde el CPWalker, como datos de entrada para los movimientos del avatar en el entorno de realidad virtual. En los futuros trabajos, se puede considerar el empleo del resto de parámetros, como las fuerzas de interacción o el ángulo de los tobillos, para obtener un sistema de interacción y sincronización completo entre el entorno de realidad virtual y la terapia asistida por robótica.

La integración de las gafas de *Oculus Rift* junto con el entorno virtual desarrollado, el cual estaba previsto implementarlo, pero por cuestión de tiempo no ha podido llevarse a cabo. El uso de las gafas de realidad virtual permite una mejor inmersión del usuario, en lugar de visualizar una representación gráfica del escenario en modo 2D. Para ello, se plantea un estudio preliminar en forma de cuestionarios en el que se evalúa la aceptación de los niños que padecen parálisis cerebral, y la recomendación de los expertos sobre el tiempo de uso de estas. El siguiente paso será el uso de las tecnologías de realidad aumentada. En la realidad aumentada, se podrá fusionar el mundo real con el virtual. Esto toma todas las ventajas de la realidad virtual y las pone al alcance del paciente, llevando la inmersión al máximo.

Además del entorno virtual, sería interesante desarrollar escenarios de historias virtuales en las que el niño interprete un personaje dentro de la historia y realice misiones. Por ejemplo, el paciente tiene que ayudar a encontrar objetos perdidos siguiendo unas pistas, de manera que el niño puede realizar los ejercicios mientras progresa y se divierte dentro del entorno virtual. La progresión y el desenlace de la historia se ajustarán a la actuación del niño al caminar, y se conseguirá un final feliz de la historia si el niño consigue alcanzar y mantener los objetivos de la rehabilitación. También se podría considerar un sistema de rehabilitación multiusuario en el entorno virtual. El objetivo de este sistema es crear una comunidad de pacientes con parálisis cerebral y que puedan interactuar entre ellos mientras realizan su rutina de ejercicios. Esta funcionalidad permite incrementar la motivación del sujeto al incorporar el componente de interacción social y de alguna manera aumenta la competitividad sana y participación.

Junto con las mejoras comentadas, sería interesante desarrollar un sistema de bases de datos del entorno virtual con el fin de llevar a cabo una monitorización y seguimiento del paciente en todo momento. Para ello habría que crear un sistema de registro en forma de formularios para iniciar sesión y registrar los datos del paciente. En principio, dicha interfaz será manejada por los profesionales sanitarios ya que son encargados de ajustar los parámetros según cada paciente. De esta forma se guardan tanto los datos de los ejercicios realizados como los parámetros seleccionados en cada sesión. Desde el entorno virtual se conocen las posiciones de cada articulación durante la sesión de rehabilitación, por lo que también se puede estudiar la velocidad con la que se mueve cada articulación, puesto que se conocen el espacio recorrido y el tiempo que tarda.

Por otra parte, la monitorización y el tratamiento de los pacientes con parálisis cerebral dependen de instalaciones de los entornos hospitalarios que requieren la presencia conjunta del médico y el paciente, lo que limita el acceso al tratamiento y seguimiento especializados para pacientes que viven en zonas rurales y remotas. El beneficio de emplear las tecnologías de realidad virtual no solo se limita a su uso en entornos hospitalarios, sino que pueden ser implementadas junto con los sistemas de telemedicina o teleasistencia. Es más, el hecho de recibir los tratamientos correspondientes a domicilio no solo supone un ahorro en el coste de tiempo o desplazamiento, sino que los niños podrán sentirse más cómodos en su propio hogar y lograr así mejores resultados de rehabilitación. Sin embargo, este planteamiento requiere la disponibilidad de dispositivos robóticos como el CPWalker así como las herramientas necesarias para reproducir el entorno de realidad virtual. Por lo que presenta dificultades de llevarse a cabo a corto plazo.

A lo largo de este trabajo se ha hecho énfasis sobre las posibilidades que ofrece las tecnologías de realidad virtual en la rehabilitación física de los pacientes, donde puede favorecer la motivación y participación de los niños, haciendo que puedan entrenar en un entorno lúdico. Sin embargo, el nivel actual de pruebas de efectividad de estas técnicas en la rehabilitación es aún limitado. Por ello, las futuras validaciones clínicas deberían tener esto en cuenta y establecer métodos y procedimientos con el objetivo de proporcionar resultados para entender y justificar el uso de las tecnologías de realidad virtual en la rehabilitación de niños con parálisis cerebral.

## VI. BIBLIOGRAFÍA

- [1] Weitzman, M. (2005). Terapias de Rehabilitación en Niños con o en riesgo de Parálisis Cerebral. *Revista Pediatría Electrónica*, 2(3), 47–51.
- [2] Centers for disease control and Prevention. (2021). Data and Statistics for Cerebral Palsy. <https://www.cdc.gov/ncbddd/cp/data.html> [Online; accessed 30-03-2021].
- [3] Beckung, E., & Hagberg, G. (2002). Neuroimpairments, activity limitations, and participation restrictions in children with cerebral palsy. *Developmental Medicine and Child Neurology*, 44(5), 309–316.
- [4] Narbona, J., & Sanchez-Carpintero, R. (1999). Parálisis cerebral infantil. *Pediatría Integral*, 3(4), 413–424
- [5] Neuropediatra. (2015). Tipos de parálisis cerebral infantil. <https://neuropediatra.org/2015/03/04/tipos-de-paralisis-cerebral-infantil/> [Online; accessed 30-03-2021].
- [6] Overview of Four Functional Classification Systems Commonly Used in Cerebral Palsy. (2017). *Children*, 4(4), 30.
- [7] Alliance, C. P. (2021). Gross Motor Function Classification System (GMFCS) | Cerebral Palsy Alliance. Cerebral Palsy Alliance. <https://cerebralpalsy.org.au/our-research/about-cerebral-palsy/what-is-cerebral-palsy/severity-of-cerebral-palsy/gross-motor-function-classification-system/>
- [8] Cerebral Palsy: Hope Through Research | National Institute of Neurological Disorders and Stroke. (2021). National Institute of Neurological Disorders and Stroke. <https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Hope-Through-Research/Cerebral-Palsy-Hope-Through-Research> [Online; accessed 30-03-2021].
- [9] Cerebral Palsy - OrthoInfo - AAOS. (2021). OrthoInfo. <https://orthoinfo.aaos.org/en/diseases--conditions/cerebral-palsy> [Online; accessed 30-04-2021].
- [10] Prieto, J., Bayona, E., & León-Sarmiento, F. (2011). Neuroplasticidad, Neuromodulación y. *Scielo.Org.Co*, 27(1), 95–107. <http://www.scielo.org.co/pdf/sun/v27n1/v27n1a10.pdf>
- [11] Cramer, S. C., Sur, M., Dobkin, B. H., O'Brien, C., Sanger, T. D., Trojanowski, J. Q., Rumsey, J. M., Hicks, R., Cameron, J., Chen, D., Chen, W. G., Cohen, L. G., Decharms, C., Duffy, C. J., Eden, G. F., Fetz, E. E., Filart, R., Freund, M., Grant, S.



- J.,...Vinogradov, S. (2011). Harnessing neuroplasticity for clinical applications. *Brain*, 134(6), 1591–1609.
- [12] Meyer-Heim, A., Ammann-Reiffer, C., Schmartz, A., Schäfer, J., Sennhauser, F. H., Heinen, F., Knecht, B., Dabrowski, E., & Borggraefe, I. (2009). Improvement of walking abilities after robotic-assisted locomotion training in children with cerebral palsy. *Archives of Disease in Childhood*, 94(8), 615–620.
- [13] Balcı, N. Ç. (2016). Current Rehabilitation Methods for Cerebral Palsy. IntechOpen. <https://www.intechopen.com/books/cerebral-palsy-current-steps/current-rehabilitation-methods-for-cerebral-palsy>
- [14] Dollar, A. M., & Herr, H. (2007). Active orthoses for the lower-limbs: Challenges and state of the art. 2007 IEEE 10th International Conference on Rehabilitation Robotics, ICORR'07, 1(c), 968–977.
- [15] Bayon, C., & Raya, R. (2016). Robotic Therapies for Children with Cerebral Palsy: A Systematic Review. *Translational Biomedicine*, 7(1).
- [16] Bayón, C., Ramírez, O., Serrano, J. I., Castillo, M. D. D., Pérez-Somarriba, A., Belda-Lois, J. M., Martínez-Caballero, I., Lerma-Lara, S., Cifuentes, C., Frizera, A., & Rocon, E. (2017). Development and evaluation of a novel robotic platform for gait rehabilitation in patients with Cerebral Palsy: CPWalker. *Robotics and Autonomous Systems*, 91, 101–114.
- [17] Aycardi, L. F., Cifuentes, C. A., Múnera, M., Bayón, C., Ramírez, O., Lerma, S., Frizera, A., & Rocon, E. (2019). Evaluation of biomechanical gait parameters of patients with Cerebral Palsy at three different levels of gait assistance using the CPWalker. *Journal of NeuroEngineering and Rehabilitation*, 16(1), 1–9.
- [18] Kaelin-Lane, A., Sawaki, L., & Cohen, L. G. (2005). Role of voluntary drive in encoding and elementary motor memory. *Journal of Neurophysiology*, 93(2), 1099–1103.
- [19] Holden, M. K. (2005). Virtual environments for motor rehabilitation: Review. *Cyberpsychology and Behavior*, 8(3), 187–211.
- [20] Wang, M., & Reid, D. (2011). Virtual reality in pediatric neurorehabilitation: attention deficit hyperactivity disorder, autism, and cerebral palsy. *Neuroepidemiology*, 36(1), 2–18.
- [21] Brütsch, K., Koenig, A., Zimmerli, L., Mérillat-Koeneke, S., Riener, R., Jäncke, L., van Hedel, H. J., & Meyer-Heim, A. (2011). Virtual reality for enhancement of robot-assisted gait training in children with central gait disorders. *Journal of rehabilitation medicine*, 43(6), 493–499.



- [22] Unity Technologies. (2021). Unity - Manual: <https://docs.unity3d.com/540/Documentation/Manual/> [Online; accessed 30-05-2021].
- [23] K. (2021). UdpClient Clase (System.Net.Sockets). Microsoft Docs. <https://docs.microsoft.com/es/dotnet/api/system.net.sockets.udpclient?view=net-5.0> [Online; accessed 30-05-2021].
- [24] Technologies, U. (2021a). Unity - Manual: Joints. Reference System. <https://docs.unity3d.com/Manual/Joints.html> [Online; accessed 30-05-2021].
- [25] Technologies, U. (2021c). Unity - Manual: Physics articulations. Unity Documentation. <https://docs.unity3d.com/Manual/physics-articulations.html> [Online; accessed 30-05-2021].
- [26] Technologies, U. (2021a). Unity - Manual. Rotation and Orientation in Unity. <https://docs.unity3d.com/2019.4/Documentation/Manual/QuaternionAndEulerRotationsInUnity.html> [Online; accessed 30-05-2021].
- [27] Introduction to Biomechanics. (2021). Anatomy Planes and Axis. <https://courses.fortlewis.edu/courses/17334/pages/introduction-to-biomechanics> [Online; accessed 30-05-2021].
- [28] Waste from Electrical and Electronic Equipment (WEEE). (2021). European Commission Environment Waste and Recycling. [https://ec.europa.eu/environment/topics/waste-and-recycling/waste-electrical-and-electronic-equipment-weee\\_es](https://ec.europa.eu/environment/topics/waste-and-recycling/waste-electrical-and-electronic-equipment-weee_es) [Online; accessed 30-05-2021].
- [29] Moran, M., Moran, M., Moran, M., Moran, M., M., Moran, M., & Moran, M. (2021). Salud. Desarrollo Sostenible. <https://www.un.org/sustainabledevelopment/es/health/> [Online; accessed 30-05-2021].
- [30] ConSalud.es. El gasto sanitario de las familias con hijos con discapacidad supera los 800 euros al mes. Consalud. [https://www.consalud.es/pacientes/terapias/el-gasto-sanitario-de-las-familias-con-hijos-con-discapacidad-supera-los-800-euros-al-mes\\_46284\\_102.html](https://www.consalud.es/pacientes/terapias/el-gasto-sanitario-de-las-familias-con-hijos-con-discapacidad-supera-los-800-euros-al-mes_46284_102.html) [Online; accessed 30-05-2021].

## ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES

### 1. IMPACTO MEDIOAMBIENTAL

Este proyecto no tiene un impacto medioambiental en sí mismo, puesto que durante su desarrollo no se generó residuos. El único coste a tener en cuenta es la energía utilizada por los dispositivos electrónicos durante la ejecución del proyecto, pero puede ser despreciable ya que cuenta con una media de 2,2 kWh diarios (8h de uso aproximadamente). Lo que sí se debe tener en cuenta es el deterioro de los equipos electrónicos utilizados (el ordenador, el CPWalker y sus componentes). Una vez que han llegado al fin de su vida útil, deben ser retirados como indica la Directiva de Residuos de Aparatos Eléctricos y Electrónicos (Waste Electrical and Electronic Equipment Directive) [28], la cual establece los criterios de retirada y reciclaje.

### 2. IMPACTO ÉTICO, SOCIAL Y ECONÓMICO

El desarrollo sostenible es aquel que permite satisfacer las necesidades presentes sin comprometer los recursos de las generaciones futuras. Actualmente, estos objetivos se encuentran clasificados en los ODS (Objetivos de Desarrollo Sostenible) que fueron creados en 2015 tras un acuerdo entre las grandes potencias en el marco de la ONU. El presente TFG se relacionaría con el objetivo número tres, que consiste en garantizar una vida sana y promover el bienestar en todas las edades con el fin de construir una sociedad próspera [29].

En España se estima que 2 de cada 1.000 personas nacidas tienen parálisis cerebral. Es decir, hay 120.000 personas en España con parálisis cerebral. Además de la repercusión emocional, los padres con hijos afectados tienden a dedicar mayor parte del tiempo al cuidado y atención. Debido a ello, la mayoría de ellos tienen que trabajar a tiempo parcial, y una gran parte del ingreso se destina a gastos en salud. En España, este valor llega a superar los 800€ al mes [30].

Por ello, este TFG tiene como objetivo estudiar el empleo de las tecnologías de RV como complemento a la rehabilitación de marcha asistida por robótica en los niños con parálisis cerebral. Tratando de incrementar su motivación, proveer nuevas formas de asistencia, y en general, mejorar el proceso global de su rehabilitación.

El objetivo del CPWalker es mejorar la calidad de vida y la autonomía de los pacientes con parálisis cerebral mediante la realización de terapias robóticas tempranas y novedosas para evitar el deterioro de las capacidades motrices. Estas terapias durante las etapas tempranas de desarrollo de los niños permiten una reducción de las deficiencias producidas por la parálisis cerebral, aumentando la calidad de vida y reduciendo los costes. El uso de la tecnología virtual complementa a la robótica en el campo de la rehabilitación haciendo que ésta última sea más atractiva. Por ello, es importante realizar estudios que analicen el uso de ambas

tecnologías con la finalidad de ofrecer una mayor accesibilidad a las personas, ya que no todas las personas pueden afrontar el coste que conlleva el uso de estas tecnologías.

La introducción de la robótica en el campo de la rehabilitación puede resultar un tema polémico cuando un robot asume parte o todo el trabajo humano. Sin embargo, hay que destacar que tanto el empleo de los dispositivos robóticos como las tecnologías de realidad virtual tienen como objetivo asistir al terapeuta y no sustituirlo, lo que les permitiría centrarse en auxiliar al paciente en otras variables para perfeccionar el movimiento y aumentar el rendimiento. Mientras que para los pacientes será un incentivo en la realización de las actividades propuestas.

Finalmente, teniendo en cuenta el impacto de la PCI en la salud mundial y el objetivo número tres de los ODS, se espera que, el presente trabajo desarrollado pueda ser de ayuda en el estudio del empleo de las tecnologías en la mejora de la salud de los pacientes con parálisis cerebral.

## ANEXO B: PRESUPUESTO ECONÓMICO

Este Trabajo de Fin de Grado ha sido realizado en colaboración con el Centro de Automática y Robótica del CSIC-UPM que ha aportado los recursos necesarios para llevar a cabo este proyecto.

Para la elaboración del presupuesto del proyecto se ha realizado un desglose de los costes directos asociados al mismo. Se consideran costes directos a aquellos costes imputables directamente al proyecto. En este caso son el coste de la mano de obra, los recursos materiales (tanto herramientas de hardware como las de software) y los gastos de amortización.

- Costes asociados a los recursos humanos: se ha calculado a partir del salario promedio (10€/h) de un estudiante de ingeniería en prácticas.
- Costes asociados a los recursos informáticos y materiales: se tendrán en cuenta la amortización de los equipos de hardware y la de licencias de software informático.
- Para el cálculo de los gastos de amortización, se ha tenido en cuenta los años de vida útil y se calcula dividiendo el precio de adquisición entre la vida útil estimada.

Recursos humanos				
Concepto	Horas	€/hora	Coste (€)	Total (€)
Estudiante de ingeniería	420	10	4.200	4.200

Tabla 1. Costes asociados con la mano de obra.

Costes del equipamiento técnico y software						
Descripción	Coste (€/unidad)	Tiempo de vida (años)	U.	Coste de amortización (€/año)	Tiempo (meses)	Total (€)
El robot CPWalker	50.000	10	1	5.000	6	2.500
SparkFun ESP32 Thing	28,5	2	1	14,25	6	7,13
Ordenador portátil Asus	1.000	5	1	200	6	100
Unity 3D	399	1	1	399	6	199,5
JetBrains Rider	139	1	1	139	6	69,5
MATLAB	800	1	1	800	2	133,3
Arduino	0	0	1	0	2	0

Tabla 2. Costes del equipamiento técnico y software.

Finalmente, el coste total estimado del proyecto se resume en la siguiente tabla:

Presupuesto final	
Concepto	Coste (€)
Costes de equipamiento	3.009,4
Recursos humanos	4.200
IVA (21%)	1.513,97
<b>Total</b>	<b>8.723,37</b>

Tabla 3. Presupuesto final del proyecto.