



# **DESARROLLO DE LA FUNCIONALIDAD DE GESTIÓN PARA TERAPEUTAS EN UN SISTEMA DE REALIDAD VIRTUAL PARA REHABILITACIÓN MOTORA**

**Rafael Moraga Gómez**

**Tutor: Beatriz Rey Solaz**

**Cotutor: José María Monzó Ferrer**

Trabajo Fin de Máster presentado en la Escuela Técnica Superior de Ingeniería de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Máster en Ingeniería Telecomunicación

Curso 2020-21

Valencia, 6 de Septiembre de 2021



## Resumen

La mala recuperación de una lesión, o posturas incorrectas del cuerpo, pueden afectar a la salud física de las personas. Con la pandemia vivida de COVID-19 a nivel mundial, y con las situaciones de confinamiento y de restricciones, la práctica de ejercicio o de rehabilitación física han supuesto un reto tanto para las personas a la hora de realizar los diferentes ejercicios necesarios para la recuperación de cualquier tipo de lesión, como de los profesionales responsables del seguimiento de estos ejercicios. El simple hecho de poder acudir a un centro de rehabilitación se ha hecho imposible, por lo que la práctica de los ejercicios necesarios, también.

Además, existe un gran número de personas que viven en zonas rurales o alejadas de cualquier centro de rehabilitación y la asistencia frecuente a estos centros no es algo viable, ya que se necesitan recorrer largas distancias hasta estos centros.

Con esta situación, han proliferado numerosos sistemas tecnológicos que han intentado paliar de la mejor manera estos problemas. Podemos considerar que una de las ramas que ha vivido su mejor momento con esta situación es la realidad virtual. La realidad virtual ha ayudado en gran parte a lidiar con diferentes situaciones en las que bien, la distancia a los centros de rehabilitación es demasiado grande o incluso es imposible acudir directamente a ellos de manera frecuente.

En este proyecto, se realiza el desarrollo de una aplicación para la definición de ejercicios de rehabilitación por parte de los fisioterapeutas. Esta aplicación proporciona al fisioterapeuta las herramientas necesarias para definir de forma sencilla e intuitiva los diferentes ejercicios que componen la sesión de entrenamiento. La aplicación es compatible con el módulo del paciente desarrollado en un trabajo fin de máster anterior, de tal manera que los ejercicios definidos por el fisioterapeuta puedan ser ejecutados por el paciente sin ningún problema. Además, se ha desarrollado un segundo módulo para poder validar los ejercicios que se han definido por el usuario de tal modo que el fisioterapeuta pueda comprobar que los ejercicios que se han definido mediante el primer módulo se han definido de forma correcta y corresponden con los ejercicios deseados para la sesión de entrenamiento. La aplicación será desarrollada para ejecutarse en un ordenador, haciendo uso de teclado y ratón de forma sencilla.



## Resum

Una dolenta recuperació d'una lesió o de postures incorrectes del cos pot afectar a la salut física de les persones. Amb la pandèmia viscuda de COVID-19 a nivell mundial, i amb les situacions de confinament i restriccions, la pràctica de exercici o de rehabilitació física han suposat un repte tant per a les persones a l'hora de realitzar els diferents exercicis necessaris per a la recuperació de qualsevol tipus de lesió, com per als professionals responsables del seguiment d'aquests exercicis. El simple fet de poder anar a un centre de rehabilitació s'ha fet impossible i per tant la pràctica dels exercicis necessaris, també.

A més a més, existeix un gran nombre de persones que viuen en zones rurals llunyanes de qualsevol centre de rehabilitació i assistir freqüentment a aquests centres no és viable, ja que necessiten recórrer llargues distàncies fins aquests centres.

Amb aquesta situació, han proliferat nombrosos sistemes tecnològics que han tractat de pal·liar de la millor forma possible aquests problemes. Podem considerar que una de les branques que ha viscut el seu millor moment amb aquesta situació és la realitat virtual. La realitat virtual ha ajudat en gran part a tractar amb diferents situacions en les quals o bé la distància als centres de rehabilitació és massa gran o de fet és impossible acudir directament a ells de manera freqüent.

En aquest projecte, es realitza el desenvolupament d'una aplicació per a la definició de exercicis de rehabilitació per part dels fisioterapeutes. Aquesta aplicació proporciona al fisioterapeuta les eines necessàries per a definir de forma senzilla e intuïtiva els diferents exercicis que componen la sessió de entrenament. La aplicació és compatible amb el mòdul del pacient desenvolupat en un treball fi de màster anterior, de manera que els exercicis que han sigut definits pel fisioterapeuta poden ser executats pel pacient sense cap problema. A més a més, s'ha desenvolupat un segon mòdul pera poder validar el exercicis per l'usuari de tal mode que el fisioterapeuta pot comprovar que els exercicis que han sigut definits en el primer mòdul han sigut definits de manera correcta i corresponen amb els exercicis desitjats per a la sessió d'entrenament. La aplicació serà desenvolupada per a que siga executada en un ordinador, mitjançant l'ús de teclat i ratolí de forma senzilla.



## Abstract

Poor recovery from an injury, or incorrect body postures, can affect people's physical health. With the COVID-19 pandemic, and with situations of confinement and restrictions, the practice of exercise or physical rehabilitation has been a challenge for people performing exercises necessary for recovery and also for the professionals responsible for following these exercises. The simple fact of being able to go to a rehabilitation center has become impossible.

In addition, there are a large number of people who lives in rural areas or far from any rehabilitation center and frequent attendance at these centers is not feasible, since long distances need to be traveled to these centers.

With the situation, numerous technological systems have proliferated trying to solve these problems in the best way possible. We can consider that one of the branches that has experienced its best moment with this situation is virtual reality. Virtual reality has helped to deal with different situations where the distance to rehabilitation center is too long or it is even impossible to go there.

In this project, an application for the definition of rehabilitation exercises by physiotherapists it is developed. This application provides the physiotherapist with the necessary tools to define in a simple and intuitive way the different exercises that make up the training session. The application is compatible with the patient module developed in a previous master's thesis, where the exercises defined by the physiotherapists can be performed by the patient without any problem. In addition, a second module has been developed to be able to validate the exercises that have been defined by the user in such a way that the physiotherapist can verify that the exercises that have been defined through the first module have been defined correctly and correspond to the exercises desired for the training session. The application will be developed to run on a computer, using the keyboard and mouse in a simple way.



## Índice

Capítulo 1.	Introducción .....	3
Capítulo 2.	Objetivos .....	4
Capítulo 3.	Metodología .....	5
3.1	Gestión del proyecto.....	5
3.2	Distribución de tareas.....	5
3.2.1	Estudio del proyecto anterior .....	5
3.2.2	Estudio sistemas de realidad virtual .....	5
3.2.3	Formación en motor gráfico Unity .....	5
3.2.4	Estudio de cinemática inversa .....	6
3.2.5	Estudio del avatar a utilizar .....	7
3.2.6	Definición aplicación para creación de ejercicios .....	7
3.2.7	Definición aplicación para validación de ejercicios .....	8
3.2.8	Desarrollo del software .....	8
3.2.9	Elaboración de la memoria.....	8
3.3	Diagrama temporal.....	8
Capítulo 4.	Estado del arte .....	1
4.1	Realidad virtual .....	1
4.2	Nuevas tecnologías aplicadas a rehabilitación .....	3
4.2.1	Rehabilitation using Kinect-based Games and Virtual Reality (ReaKinG) .....	3
4.2.2	Kinect-based telerehabilitation system.....	4
4.2.3	ReHabGame. Un sistema no inmersivo de rehabilitación con aplicación en neurociencia .....	5
4.3	Descripción de la aplicación del paciente .....	6
Capítulo 5.	Desarrollo y resultados.....	9
5.1	Desarrollo.....	9
5.1.1	Creación del avatar.....	9
5.1.2	Implementación cinemática inversa .....	11
5.1.3	Desarrollo matriz de esferas .....	15
5.1.4	Implementación sistema de colisiones .....	16
5.1.5	Implementación sistema definición de ejercicios.....	20
5.1.6	Desarrollo escritura archivo JSON.....	24
5.1.7	Desarrollo aplicación validación de ejercicios .....	25
5.2	Resultados aplicación para definición de ejercicios.....	29



5.3	Resultados aplicación para validación de ejercicios .....	31
Capítulo 6.	Conclusiones y propuesta de trabajo futuro .....	33
6.1	Trabajo futuro.....	33
Capítulo 7.	Bibliografía.....	36

## Capítulo 1. Introducción

El periodo de confinamiento que supuso la pandemia de COVID-19 vivida a finales del año 2019, dejó en aislamiento social a la gran mayoría de la población mundial, generando una pérdida de rutina diaria y de actividad física asociada que esta conlleva. En este grupo poblacional, podemos encontrar personas mayores y pacientes con lesiones físicas. La fisioterapia para estos grupos sociales estaría orientada a mantener unas rutinas de actividad física y/o ejercicio de rehabilitación adaptado a cada paciente. Siempre que sea posible, se realizará esta actividad de una manera telemática intentando evitar en la mayor medida de lo posible el contacto social para evitar la probabilidad de contagio del virus. Así, durante estos últimos meses, han surgido numerosos métodos para poder llevar a cabo esta terapia o seguimiento de manera remota, como vídeos, llamadas de seguimiento, etc.

Además, existe un grupo de personas que viven en zonas rurales o alejadas de los hospitales y centros de rehabilitación. El acceso de estas personas de forma frecuente a estos centros, resulta una actividad poco viable en muchas ocasiones, donde el centro más cercano puede situarse a más de 100 km, por lo que una rehabilitación de forma telemática resultaría muy interesante en estos casos.

Este proyecto se basa sobre un trabajo fin de máster anterior, de Yari Mirko Anoffo, (2019) *“Development of a virtual environment for full-body motor training”*, donde se desarrolla una aplicación para pacientes de los grupos sociales anteriormente mencionados, de tal manera que se proporciona una manera de llevar a cabo la terapia de rehabilitación física de forma totalmente telemática, mediante el uso de un sistema de realidad virtual. Este trabajo, estaba orientado al paciente, de tal modo que los ejercicios ya se encontraban definidos y no era posible definirlos desde la aplicación.

Por ello, surge necesidad de desarrollo del proyecto descrito en el presente documento, una aplicación que proporcione al fisioterapeuta las herramientas necesarias para poder definir los ejercicios de manera telemática para la aplicación del paciente. La aplicación constará de 2 módulos diferenciados, uno de ellos dirigido a la definición de los ejercicios, y un segundo que permita el proceso de validación de ejercicios.

Gracias a esta aplicación, el fisioterapeuta podrá realizar de una manera telemática, la rutina de rehabilitación de sus pacientes, proporcionando claras ventajas como ahorro de desplazamientos, contacto social e incluso una mayor comodidad para ambas partes. Además, esta aplicación supondrá un ahorro de tiempo para el fisioterapeuta, que se traducirá en la posibilidad de tratar a un mayor número de pacientes de forma más eficiente, lo que supondrá un mayor beneficio económico para el fisioterapeuta.

El documento está organizado en 5 capítulos diferentes:

- Una vista sobre el proyecto y sus objetivos.
- La metodología seguida para su desarrollo y descripción de tareas que lo componen
- Estado del arte
- Desarrollo del proyecto y los resultados obtenidos
- Conclusiones y posible trabajo futuro



## Capítulo 2. Objetivos

El principal objetivo de este trabajo es solucionar uno de los principales problemas con los que se han encontrados los fisioterapeutas, definir a distancia los ejercicios que sus pacientes deben de llevar a cabo para realizar la rehabilitación para una lesión sufrida. Además, es útil también para pacientes que se encuentren alejados de las clínicas de rehabilitación, o tengan cualquier problema para desplazarse hasta ellas.

El trabajo se basa sobre uno anterior, donde se desarrolló un sistema de realidad virtual que proporcionaba al paciente mediante el uso de un visor de realidad y virtual y una cámara Kinect, un sistema inmersivo para rehabilitación de lesiones. Este sistema desarrollado para el paciente utiliza un fichero JSON para generar los ejercicios necesarios. Generar este fichero con la estructura necesaria, no es algo cómodo para un fisioterapeuta, el cual no tiene por qué conocer la estructura de este tipo de ficheros, además incluye numerosos parámetros difíciles de definir de forma manual, por lo que es necesario la existencia de una aplicación que permita generar esta estructura y tipo de datos sin tener conocimiento de ella, de forma sencilla e intuitiva.

Para este trabajo, se desarrolla la parte del fisioterapeuta, encargada de definir los diferentes ejercicios que posteriormente serán cargados en la aplicación del cliente para su realización.

Los objetivos concretos que se pueden proponer al a partir del objetivo principal son:

- Desarrollar la aplicación para poder ser ejecutada en un PC con sus periféricos habituales como teclado y ratón. De esta manera, se facilita el acceso a la aplicación a un mayor número de terapeutas.
- Proporcionar a los terapeutas la posibilidad de generar sesiones de entrenamiento para los pacientes de forma remota.
- Mantener la compatibilidad con la aplicación del cliente, para ello, la aplicación debe generar un fichero JSON con la misma estructura que utiliza la aplicación del cliente.
- Desarrollar un módulo secundario que permita al terapeuta poder validar los diferentes ejercicios que se definan en la aplicación de una manera sencilla para verificar los ejercicios de la sesión de entrenamiento.
- Posibilitar la sencilla adaptación de la aplicación para poder ser ejecutada en otros sistemas, como pueden ser sistemas de realidad virtual, smartphones, tablets...etc.



## Capítulo 3. Metodología

En este capítulo, se define la metodología que se ha seguido para el desarrollo del proyecto, indicando las diferentes tareas en las que se ha subdividido el trabajo, además, se incluye un diagrama temporal en el que se proyecta el tiempo dedicado a cada subtarea y la distribución temporal del proyecto.

### 3.1 Gestión del proyecto

El proyecto se ha subdividido en diferentes tareas para facilitar el desarrollo del mismo. A lo largo del proyecto, se han mantenido reuniones periódicas con los tutores para realizar un seguimiento y poder solucionar los problemas que han ido surgiendo, además de redefinir los objetivos parciales en caso de ser necesario. A continuación, se realiza una descripción detallada de cada una de las tareas que ha supuesto el proyecto, así como la manera de resolverlas y los posibles problemas que se han podido encontrar en cada una de ellas.

### 3.2 Distribución de tareas

#### 3.2.1 Estudio del proyecto anterior

En la primera parte del proyecto, se ha estudiado el proyecto anterior sobre el que se está continuando el trabajo. DEVELOPMENT OF A VIRTUAL ENVIRONMENT FOR FULL-BODY MOTOR TRAINING [1] consiste en un software para realidad virtual donde se facilita al paciente la posibilidad de realizar diferentes ejercicios para rehabilitación de lesiones físicas.

El objetivo del trabajo anterior consiste en el desarrollo de una aplicación para realidad virtual de bajo coste, donde el paciente simplemente necesita un visor y una cámara Kinect para la realización de los ejercicios.

El visor utilizado en este caso es el Oculus Rift, un visor de un coste bajo que permite al usuario una inmersión en un entorno virtual. Para el seguimiento del movimiento del paciente, en lugar de utilizar los controladores que ofrece el visor de realidad virtual, se ha optado por utilizar una cámara Kinect, que posibilita el seguimiento de todo el movimiento del cuerpo humano.

#### 3.2.2 Estudio sistemas de realidad virtual

Para la realización del trabajo, ha sido necesario realizar un estudio sobre los sistemas actuales de realidad virtual.

La realidad virtual, es una tecnología que permite crear un mundo ficticio, del que se puede formar parte e interactuar con los elementos que lo componen. Para ofrecer esta posibilidad, se hace uso normalmente de elementos como un visor y unos controladores que permitan la interacción.

#### 3.2.3 Formación en motor gráfico Unity

Para el desarrollo del proyecto, se ha escogido Unity como motor gráfico para el desarrollo de la aplicación. Unity es un motor gráfico que permite la creación de juegos y aplicaciones ya sea en 2D, 3D o incluso para realidad virtual a gran velocidad. Es uno de los motores gráficos más extendidos actualmente en el desarrollo de videojuegos, permite además la adaptación de las aplicaciones de forma sencilla a numerosas plataformas, desde videoconsolas, ordenadores, hasta incluso a dispositivos móviles Android e IOS. Es un software multiplataforma disponible para Mac, Windows o Linux.



**Figura 1. Motor gráfico Unity**

Unity ha sido la elección para la elaboración de la aplicación debido a que es un motor gráfico muy extendido, además, ofrece una licencia gratuita si eres estudiante o bien para personas individuales o empresas pequeñas si los ingresos o fondos recaudados en los 12 meses previos son inferiores a 100.000\$. Unity ofrece también, una tienda virtual de assets comerciales y gratuitos creados por la propia Unity o por miembros de la comunidad, existen una gran cantidad de assets disponibles, desde texturas, modelos y animaciones, hasta proyectos completos que pueden ser descargados y utilizados en otros proyectos.

En Unity, cada objeto es denominado *GameObject*. Estos objetos necesitan tener definidas diferentes propiedades para convertirse en un personaje, un efecto especial, animación, o cualquier cosa que pueda ser utilizada en una escena del proyecto. Estas propiedades especiales es lo que se conoce como *componentes*.

Un conjunto de *GameObjects* componen una escena. Una escena es como cada nivel en la aplicación final, son bastante útiles para subdividir el trabajo o proyecto en diferentes partes. Mediante Unity es posible animar los objetos que componen una escena, permitiendo añadir audio o efectos de video. Para interactuar con la aplicación, Unity permite también crear paneles o menús interactivos para el usuario. Para crear todas estas animaciones o interacciones dentro de la aplicación, se pueden utilizar herramientas nativas existentes en Unity o usando código programable escrito en algunos lenguajes de programación, y asociando este código a los objetos que componen la escena.

Otra de las principales características de Unity, es que ofrece la posibilidad de programar los scripts en diferentes lenguajes de programación, entre Boo, Unity Script y C#. Aunque realmente, siempre se recomienda el uso de C#, ya que es el lenguaje más ampliamente utilizado, y del que más soporte se dispone.

Dado que Unity ofrece infinitas posibilidades y complejas herramientas, ha sido necesaria una formación en el motor para poder desarrollar el software. Esta formación, se ha realizado a través de plataformas online, realizando cursos proporcionados por la Universidad Politècnica de Valencia.

#### **3.2.4 Estudio de cinemática inversa**

La cinemática, estudia el movimiento de un personaje respecto a un sistema de referencia sin considerar las fuerzas que intervienen.

El concepto de cinemática inversa es muy importante tanto para videojuegos como para animaciones en 3D. La cinemática inversa es un proceso de cálculo matemático que permite el obtener la posición final de las articulaciones de una cadena cinemática, como puede ser el

esqueleto de un personaje 3D. Esta posición puede ser calculada mediante el uso de múltiples fórmulas trigonométricas.

En este proyecto el concepto de cinemática inversa tiene gran relevancia, ya que será necesario aplicarlo para poder calcular correctamente la posición de las articulaciones del avatar que se utilizará para la definición de los ejercicios. La cinemática inversa es importante porque permite mover de forma realista las articulaciones aplicando movimiento a una única parte del cuerpo. De esta forma, desplazando únicamente por ejemplo la mano, la cinemática inversa permite predecir la posición del codo y del hombro de una manera realista. Para este proyecto esto es importante, pues nos permite definir los movimientos del brazo entero desplazando únicamente la mano. Al simular de la manera más realista posible el posible movimiento del avatar, servirá para asimilar el movimiento que tendrá que realizar el paciente posteriormente a la hora de realizar los ejercicios, y de esta manera asegurar que se han definido los ejercicios de la manera más correcta y eficiente para la rehabilitación del paciente.

### **3.2.5 Estudio del avatar a utilizar**

Una vez realizado un estudio sobre el motor gráfico Unity y los fundamentos de la cinemática inversa, se realiza un estudio del avatar a utilizar en la aplicación.

Un avatar es básicamente una representación gráfica digital asociada a un usuario para identificarlo dentro de un juego virtual. El avatar puede ser una fotografía, un icono, una figura e incluso puede tomar forma tridimensional.

El avatar será fundamental a la hora del desarrollo de la aplicación, ya que será la pieza clave del software. Será necesario para que el fisioterapeuta pueda utilizarlo como herramienta para definir los diferentes ejercicios de forma correcta, de manera que simule al paciente final. Para ello, se estudian las diferentes opciones posibles para la generación de este avatar, desde el uso de herramientas de creación de avatares (Makehuman) hasta el uso directo de la tienda de assets de Unity.

### **3.2.6 Definición aplicación para creación de ejercicios**

En esta parte del proyecto, se estudia la definición de la aplicación para la creación de los ejercicios por parte del fisioterapeuta. Es importante definir la aplicación de tal manera que sea sencilla para el usuario final. Para ello, se barajan las diferentes opciones a la hora de definir los ejercicios y del movimiento del avatar.

En una primera fase la aplicación se basa en el uso de teclado para el control del avatar, sin embargo, se estudian diferentes alternativas que puedan proporcionar otras maneras de manejo de la aplicación.

La primera alternativa de control de la aplicación es el uso de un visor de realidad virtual al igual que en la aplicación del paciente, y mediante el uso de los controladores del propio dispositivo, realizar la interacción con el software. También se estudia la utilización de dispositivos de tracking para otras partes del cuerpo como las piernas o el torso. El problema que presenta esta solución es que se necesita un dispositivo de realidad virtual y los dispositivos de tracking, lo cual supone un aumento del coste de la aplicación.

Otra alternativa estudiada para la definición de ejercicios es el uso de una cámara para el seguimiento de los movimientos del fisioterapeuta y trasladarlos al avatar. La ventaja de esta alternativa es que presenta una manera más sencilla de realizar el movimiento del avatar. Como desventaja, al igual que ocurre en el caso anterior, es necesario incluir más hardware que aumentaría el coste del producto, aunque en este caso el coste es mucho menor que un dispositivo de realidad virtual.

El uso de cualquiera de las alternativas propuestas para realizar el tracking del movimiento del usuario no supondría una modificación muy compleja del software, ya que simplemente será necesario la modificación de los scripts encargados del movimiento del avatar, el resto del

software mantendrá su funcionalidad. Por ello, se propone como trabajo futuro en apartados posteriores el uso de alguna de estas alternativas para mejorar la aplicación final.

### 3.2.7 *Definición aplicación para validación de ejercicios*

En esta parte, se estudia el módulo secundario para la validación de los ejercicios definidos con el módulo anterior. Al igual que anteriormente, es necesario que esta aplicación sea sencilla para el usuario, y de manera clara proporcione una visión sobre los ejercicios que se han definido anteriormente y permita validarlos de manera correcta.

Este módulo secundario es importante para el fisioterapeuta, ya que proporciona una herramienta para comprobar que la sesión de entrenamiento está definida de forma correcta, es realizable por el paciente y no supondrá ningún problema.

Dentro del software desarrollado, utilizará en gran medida parte del módulo anterior, ya que el funcionamiento básico es el mismo, un avatar rodeado por una matriz de esferas donde el fisioterapeuta tendrá que realizar los ejercicios simulando el comportamiento del paciente, y comprobar que todas las esferas definidas en el módulo anterior se han registrado correctamente.

### 3.2.8 *Desarrollo del software*

Tras haber realizado la definición de los dos módulos que se van a desarrollar, estudiado el motor gráfico a utilizar, y de las herramientas de cinemática inversa y avatar que se van a utilizar para el proyecto, se pasa al desarrollo del software, donde se implementa todo lo anteriormente comentado en dos aplicaciones independientes, que pueden ser incorporadas a un software conjunto en un futuro.

Para el desarrollo del software se ha hecho uso del IDE (*Integrated Development Environment*) Microsoft Visual Studio. Este IDE es compatible con numerosos lenguajes de programación, como pueden ser Java, PHP, C++, Javascript ,etc.



Figura 2. Microsoft Visual Studio

El software ha sido desarrollado en C#, ya que es uno de los lenguajes más utilizados para desarrollar en Unity. Esto es importante, ya que existe una gran comunidad desarrollando con este lenguaje y se puede encontrar de forma sencilla numerosos proyectos y soluciones para facilitar la programación de scripts.

### 3.2.9 *Elaboración de la memoria*

Finalmente, una vez desarrollado todo el software, se elabora la memoria del proyecto donde se detalla todo el proceso realizado, así como los problemas y resultados obtenidos.

## 3.3 **Diagrama temporal**

El proyecto ha tenido una duración aproximada de 6 meses, a continuación, se muestra un diagrama de Gantt con la planificación exacta de tiempos de cada una de las tareas de las que se compone el proyecto.

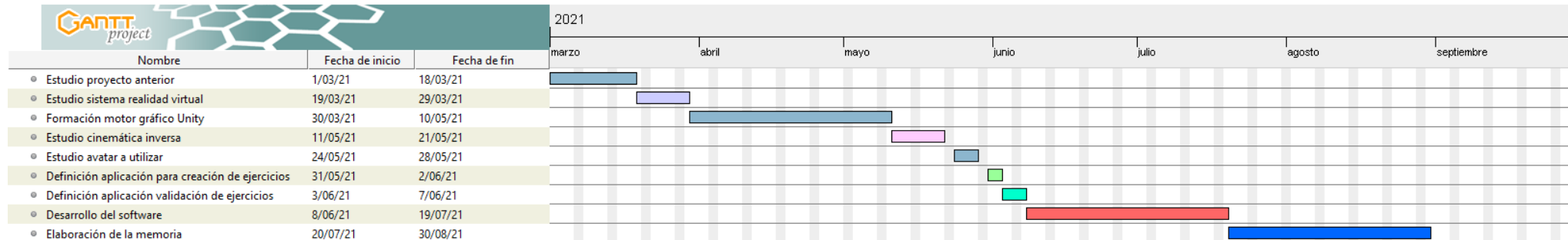


Figura 3. Diagrama de Gantt



## Capítulo 4. Estado del arte

### 4.1 Realidad virtual

Como ya se ha comentado anteriormente, la realidad virtual es una tecnología relativamente nueva, que nos permite sumergirnos en un mundo virtual e interactuar con los elementos que lo componen como si nos encontráramos dentro de él.

Esta tecnología es ampliamente usada en numerosos sectores como, por ejemplo:

- Medios de comunicación: ofreciendo al usuario videos en 360° retransmitidos en directo por streaming
- Arquitectura: ayudando a los arquitectos a proyectar sus ideas y planificar mejor el espacio
- Educación: cada día esta tecnología se usa más en las aulas, ayudando a los estudiantes a retener mejor los conocimientos y ayudando a los estudiantes con mayores dificultades
- Industria: ayudando a los operarios de fábricas a practicar y testear en entornos virtuales
- Cultura: proporcionando visitas inmersivas a museos o cines
- Medicina: ayuda en tratamientos para pacientes con ciertos problemas

Actualmente, existen numerosos visores de realidad virtual, que han ido evolucionando a lo largo de los años, mejorando tanto la calidad, como la experiencia inmersiva. La aparición de esta tecnología supone uno de los cambios tecnológicos más importantes de los últimos años. Se trata de una tecnología en pleno auge, que está consiguiendo que cada día sea más accesible para el usuario final, ya que los primeros dispositivos que empezaron a usar esta tecnología eran demasiado caros y poco accesibles al público.



**Figura 4. Dispositivo realidad virtual**

Existen distintos tipos de dispositivos de realidad virtual, normalmente conectados a un ordenador utilizando el potencial de este para funcionar. También podemos encontrar visores que hacen uso de un smartphone utilizando directamente la pantalla de este para proporcionar la experiencia. En los últimos años han proliferado también, dispositivos de realidad virtual totalmente autónomos, donde el procesamiento se realiza directamente en el CPU del que dispone el dispositivo. Aunque estos dispositivos están evolucionando rápidamente, aún presentan problemas en muchos usuarios, como el llamado efecto *motion sickness* [2], que consiste en un mareo que se produce por el desajuste existente entre nuestro cuerpo y la visión del mundo virtual. Además, los dispositivos actuales presentan también algunos problemas de resolución o nitidez, que poco a poco se están solucionando, consiguiendo incluso visores capaces de reproducir contenido en calidad 8K.

La realidad virtual es una de las tecnologías de la actualidad con mayor previsión de crecimiento. Cada día, son más las empresas que invierten en esta tecnología, desarrollando nuevas aplicaciones y dispositivos, acercando cada vez más esta tecnología al usuario.

En el entorno de trato de lesiones y rehabilitación, la realidad virtual juega un papel muy importante. En los últimos años ha surgido el término de *gamificar* [3], que consiste en aplicar estrategias usadas en juegos en contextos no jugables, con el fin de motivar a las personas a realizar una actividad y que adopten ciertos comportamientos. De esta manera, convirtiendo ejercicios de rehabilitación convencionales en un juego, tendrá efectos muy positivos en la motivación de los pacientes, permitiendo conseguir unos mejores resultados en menos tiempo y de forma más efectiva.

El uso de la realidad virtual para sesiones de rehabilitación no ofrece solamente mejores resultados en el rendimiento de los pacientes, si no que ayuda a que el experto fisioterapeuta pueda obtener unos resultados mucho más precisos y completos del rendimiento de sus pacientes.

Aunque parezca algo nuevo, el uso de realidad virtual en los centros de rehabilitación lleva realizándose muchos años [4], sin embargo, la novedad ahora es el uso de dispositivos de realidad

virtual mucho más accesibles para el usuario desde su propio hogar, en lugar de tener que desplazarse a los centros de rehabilitación, ofreciendo así una manera mucho más cómoda para el paciente, ahorrando costes y tiempos de desplazamiento, así como la comodidad de realizar las sesiones desde su propio hogar.

## 4.2 Nuevas tecnologías aplicadas a rehabilitación

En los últimos años, el campo de la realidad virtual ha crecido de forma inmensa. El uso de aplicaciones prácticas con esta tecnología es muy diverso, desde entrenamientos de aviación y aplicaciones militares hasta entrenamiento industrial en manejo de máquinas, pasando incluso por medicina, para el entrenamiento de cirujanos en operaciones complicadas.

Uno de los campos que se ha beneficiado de estas nuevas mejoras en la realidad virtual, ha sido la rehabilitación física. A continuación se describen algunas de las aplicaciones que han surgido para rehabilitación física que hacen uso de la tecnología de realidad virtual. Estas aplicaciones hacen uso de un hardware parecido al utilizado en la aplicación descrita en este proyecto.

### 4.2.1 *Rehabilitation using Kinect-based Games and Virtual Reality (ReaKinG)*

El objetivo principal de este proyecto, es la mejora de la salud de personas de avanzada edad, mediante el entrenamiento físico y mental, mientras se encuentran en su propio hogar. El sistema que se propone es *user-friendly* y está basado en la interacción natural con el cuerpo, de tal manera que las personas mayores que no están acostumbradas a utilizar ordenadores puedan disfrutar de la aplicación sin complicaciones.

La aplicación utiliza como tecnología principal el uso de una cámara Microsoft Kinect V2, que permite capturar el movimiento del usuario para interactuar con la aplicación.

ReaKinG [5] se compone de dos tipos de juegos que incluyen diferentes ejercicios para que el usuario pueda entrar o rehabilitar varios aspectos como la fuerza, o las capacidades aeróbicas y cognitivas.

Los juegos aeróbicos se componen de 4 ejercicios diferentes. El primero para entrenar el tren inferior, el segundo para el tren superior, el tercero una combinación de ambos y en el cuarto se puede entrenar cualquier parte de las extremidades con un entrenamiento cognitivo al mismo tiempo. Para las actividades, se presentan diferentes paisajes donde el paciente deberá estar presente enfrente del dispositivo Kinect y realizar una serie de movimientos detectados por el sensor. Todos los movimientos realizados son transferidos al avatar que está dentro del paisaje anteriormente mencionado. El usuario puede ver como el avatar realiza cualquier movimiento que esté haciendo ante el sensor.



Figura 5. ReaKinG



Para los juegos de fuerza, el paciente tiene que practicar algunos ejercicios como si se encontrara en un gimnasio. La plataforma explica al paciente que tipo de ejercicios debería de realizar, cuanto tiempo debe descansar y el peso que debe de utilizar. La escena del juego se sitúa en una playa, donde el paciente debe de mover un avatar con su propio cuerpo, y hay otro avatar que indica el movimiento que debe de realizar. Mientras el paciente practica los ejercicios, aparecen algunos nuevos avatares corriendo a lo largo de la playa, o incluso algunos aviones o barcos. Esto proporciona al paciente una mayor motivación y entretenimiento mientras realiza los ejercicios para mejorar los resultados.

La terapia de fuerza es diseñada por un terapeuta que puede elegir de una lista entre los diferentes ejercicios disponibles. Cada paciente podrá tener diferentes ejercicios en función de lo que el terapeuta desee evaluar.

Mientras el paciente está jugando, el sistema realiza un conteo del número de series y repeticiones realizando una recolecta de datos para procesarlos posteriormente y analizar el desempeño del paciente.

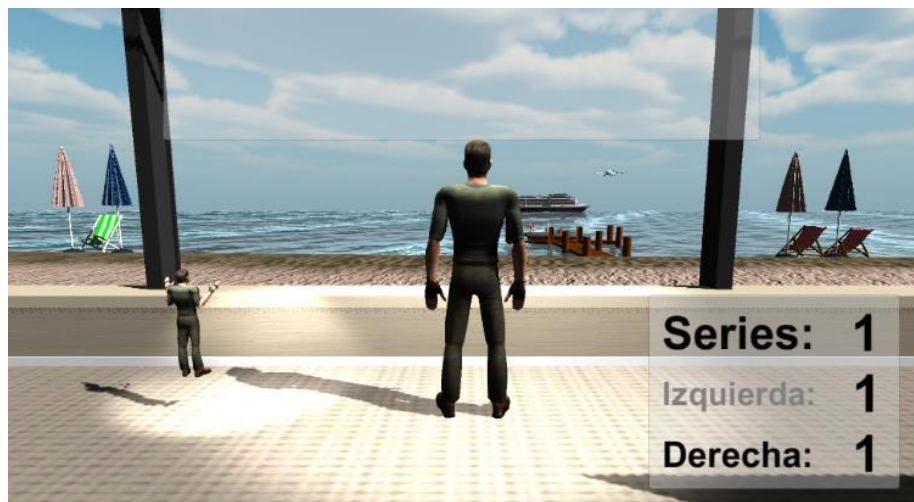


Figura 6. Sesión de fuerza ReaKinG

#### 4.2.2 *Kinect-based telerehabilitation system*

KiReS (Kinect Rehabilitation System) [6] es un sistema de rehabilitación física que permite al usuario controlar e interactuar el sistema mediante movimientos y comandos de voz. KiReS [6] proporciona una manera natural de interacción mediante el uso de 2 avatares. Estos avatares animados son utilizados para atraer la atención del usuario. Mirando al primer avatar, el usuario puede visualizar los ejercicios que debe de realizar y mirando el otro avatar, puede comprobar cómo los está realizando. Esta característica puede ayudar al usuario a corregir la ejecución del ejercicio cuando no lo está realizando de manera correcta.

Desde el punto de vista del terapeuta, KiReS [6] proporciona la posibilidad de definir las sesiones de entrenamiento que debe de realizar el paciente mediante el uso de un banco de ejercicios almacenados en el sistema. Para definir nuevos ejercicios, el terapeuta debe ejecutarlos enfrente del sistema para que pueda registrarlos. Además, el terapeuta puede analizar los datos guardados de los usuarios con el fin de redefinir los ejercicios que puedan ser problemáticos para los usuarios.

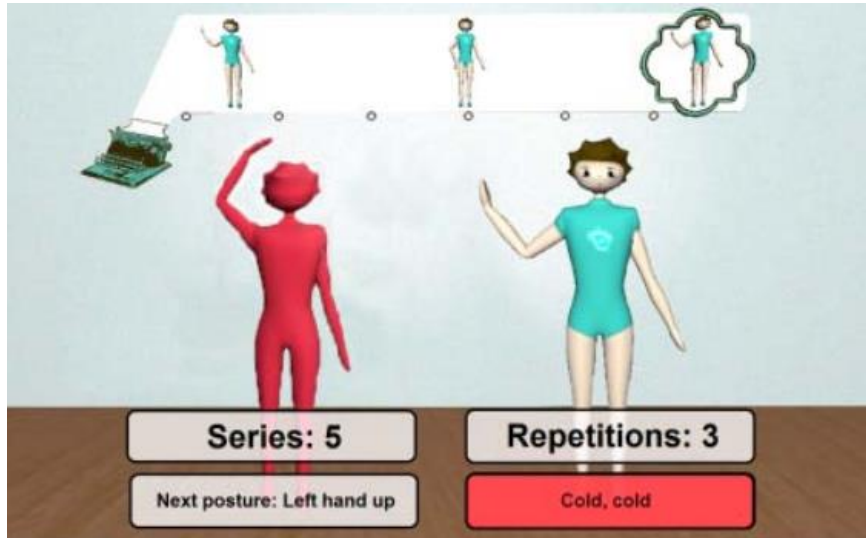


Figura 7. KiReS

Las ventajas de esta aplicación, son el uso de una interfaz muy sencilla para el usuario y el aporte de una motivación extra al paciente, la cual asegura una mejor rehabilitación y ejecución de los ejercicios.

#### 4.2.3 *ReHabGame. Un sistema no inmersivo de rehabilitación con aplicación en neurociencia*

En esta aplicación se propone el uso de un sistema de rehabilitación no-inmersivo con el uso de realidad virtual, mediante el uso de Microsoft Kinect y la pulsera por control de gestos *Thalmic Labs Myo*. ReHabGame [7] se desarrolla en base a un sistema de rehabilitación neuromuscular que ayuda al usuario a combatir las lesiones físicas y mejorar la movilidad del tren superior. Es un sistema de realidad virtual, flexible, no inmersivo y de bajo coste. Este sistema ayuda a los pacientes a interactuar con objetos virtuales que aparecen en la escena, proporcionando un fácil proceso de rehabilitación física mediante la ejecución de diferentes ejercicios. Ofrece la posibilidad de ejecutarse para un único paciente, donde el avatar del terapeuta instruye al paciente para realizar los ejercicios, o para múltiples pacientes donde los pacientes juegan al mismo tiempo compitiendo entre ellos. Todos los datos recogidos por la aplicación son almacenados y permiten posteriormente el análisis para la mejora de las sesiones de entrenamiento.

El funcionamiento de la aplicación se basa en un mundo 3D donde se generan frutas a lo largo de la escena. El usuario debe de recoger estas frutas y lanzarlas en una cesta virtual para aumentar su puntuación.



Figura 8. ReHabGame

Mediante la ejecución de estos ejercicios, se consigue el entrenamiento del tren superior del paciente, introduciendo un entorno competitivo y divertido que ayuda en la motivación del usuario para unos mejores resultados.

#### 4.3 Descripción de la aplicación del paciente

A continuación, se realiza una descripción de la aplicación desarrollada en un TFM anterior para ser ejecutada en la parte del paciente.

Esta aplicación proporciona al paciente una herramienta para realizar los ejercicios de forma remota desde su casa, proporcionando un ambiente más cómodo, así como ahorrando costes en desplazamiento y en tiempo.

La aplicación básicamente genera una matriz de esferas transparente alrededor del paciente, y a continuación aparece una esfera de color, el paciente debe de tocar dicha esfera con sus extremidades para que aparezca la siguiente esfera. El entorno virtual ofrece un espejo donde se pueden observar los movimientos que va realizando el avatar, el cual también puede ser escogido entre sexo masculino o femenino.

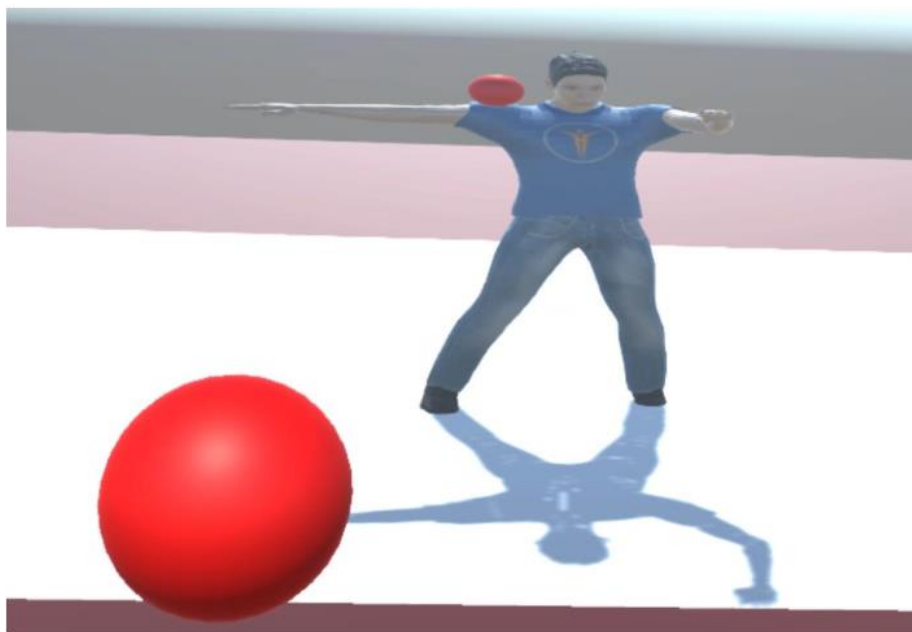


Figura 9. Aplicación Development of a virtual environment for full-body motor training

Para generar los ejercicios, el software lee un fichero JSON donde se indica la estructura de cada ejercicio, con el número de repeticiones y la posición de la esfera correspondiente. Para ello, ha sido necesario también estudiar la estructura que se ha definido para el archivo JSON, los diferentes campos que lo componen y como actúa la aplicación en función de los valores de cada uno de ellos.

A continuación, se muestra un ejemplo de un fichero de entrada JSON donde se ha definido una sesión de entrenamiento para el paciente.

```
{
  "motor_rehabilitation": [
    {
      "general_description": "This session will have two exercises! One for arms and one for legs!",
      "number_of_exercises": 2,
      "exercises": [
        {
          "exercise_description": "Exercise 1",
          "sphere_position": [213, 253, 257, 258, 266],
          "repetitions": 3,
          "user_position": "standing",
          "pause": 10
        },
        {
          "exercise_description": "Exercise 2",
          "sphere_position": [213, 281, 254],
          "repetitions": 3,
          "user_position": "standing",
          "pause": 10
        }
      ]
    },
    {
      "general_description": "This session will have one exercise! Only for arms!",
      "number_of_exercises": 1,
      "exercises": [
        {
          "exercise_description": "Exercise for arms",
          "sphere_position": [89, 91, 217],
          "repetitions": 5,
          "user_position": "standing",
          "pause": 10
        }
      ]
    }
  ]
}
```

Figura 10. Ejercicio JSON ejemplo

Para el seguimiento del movimiento del cuerpo del paciente, se hace uso de una cámara Kinect, un dispositivo compuesto por una cámara RGB, una cámara infrarrojos y un proyector láser. La cámara infrarrojos detecta patrones creados por el proyector láser para reproducir el objeto o el cuerpo. La cámara RGB añade imágenes de colores a los objetos detectados.



Figura 11. Sensor Kinect

Los principales objetivos que tenía el proyecto eran:

- Explorar técnicas de rehabilitación basadas en sistemas de realidad virtual



- Desarrollar un sistema de seguimiento completo del cuerpo humano mediante Oculus Rift
- Definir y desarrollar una aplicación para rehabilitación basada en realidad virtual

La aplicación permite la ejecución de diferentes tipos de ejercicios, es un sistema intuitivo para usuarios de todas las edades, y evita acciones complejas o interacciones con el sistema. Los ejercicios empleados para rehabilitación son definidos por un especialista en el área y tendrán seguimiento por su parte.

El proyecto incluye posibles mejoras, como la utilización de un hardware más potente para la detección del cuerpo humano, o incluso el uso de otros dispositivos para monitorizar otros parámetros como el pulso cardiaco del paciente. Estas mejoras que pueden ser implementadas, no afectarán al desarrollo del presente trabajo final de máster.

## Capítulo 5. Desarrollo y resultados

### 5.1 Desarrollo

En el presente capítulo se detalla el desarrollo del proyecto, así como los resultados obtenidos. Para ello, se detallan las diferentes fases que se han ido desarrollando, así como las soluciones implementadas.

#### 5.1.1 Creación del avatar

El primer paso del desarrollo es la creación del avatar. El avatar, será la herramienta utilizada por el fisioterapeuta para simular al paciente final, de tal manera que siga los movimientos para definir los ejercicios que el paciente debe de realizar. Para la creación del avatar se barajan diferentes opciones.

La primera de ellas consiste en la utilización de una herramienta de licencia libre denominada Makehuman. Makehuman es una herramienta que permite la creación de modelos en 3D, para posteriormente utilizarlos en juegos o animaciones. Se trata de una herramienta multiplataforma respaldada por una gran comunidad.

Permite la creación y personalización de avatares, ofreciendo la posibilidad de modificar diversos parámetros como la altura, musculatura, ropa y otros parámetros del avatar.

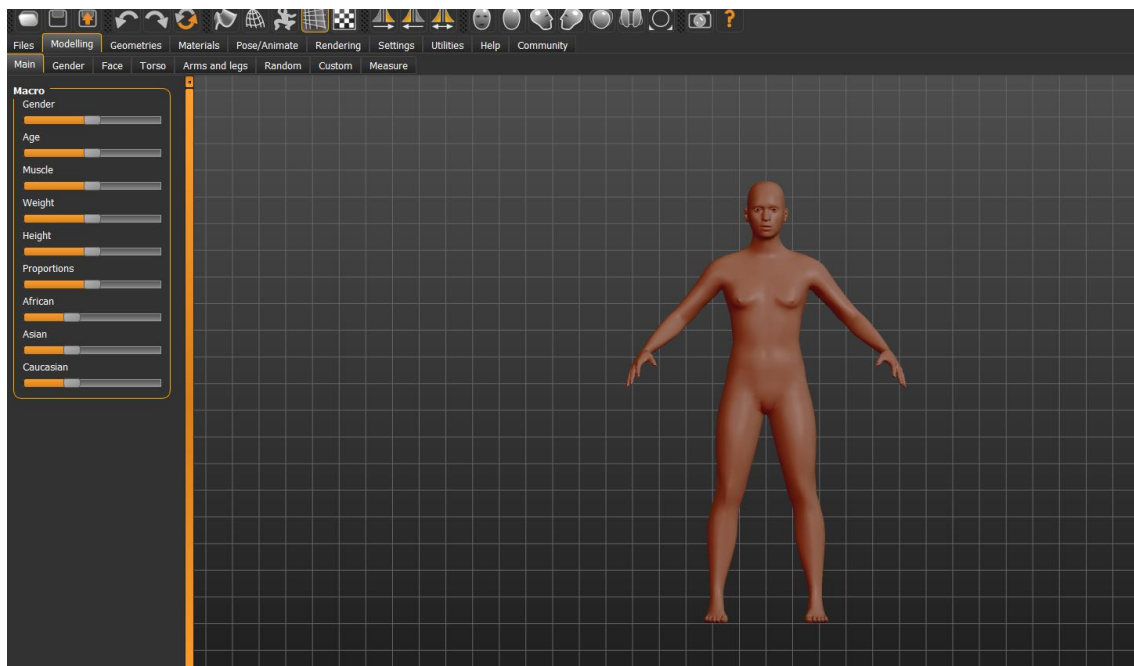


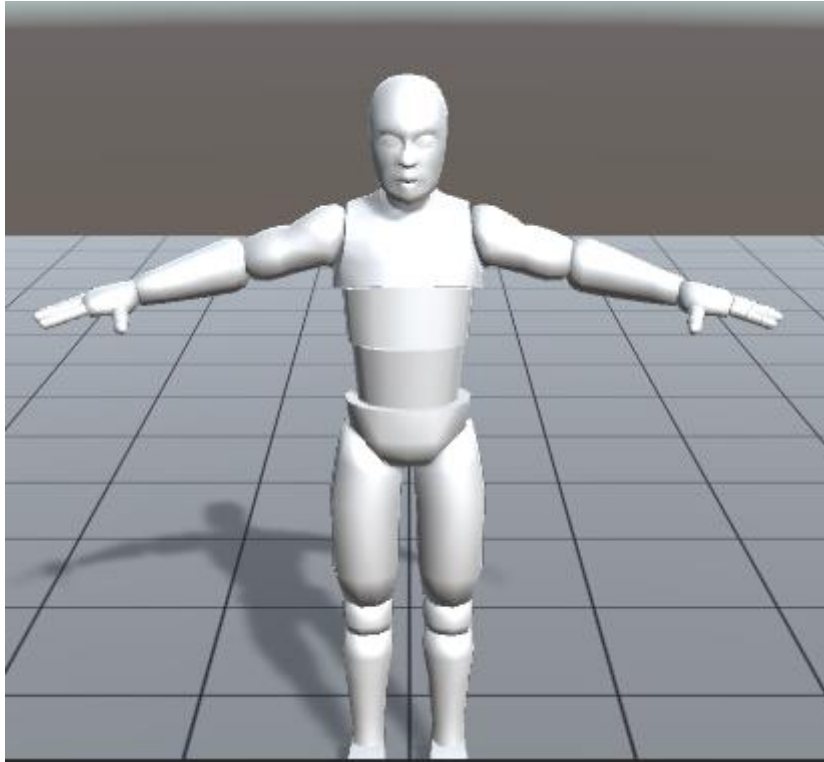
Figura 12. Creación de avatar MakeHuman

Aunque resulta una herramienta muy interesante, finalmente no se ha utilizado para el desarrollo del proyecto, ya que como se comentará en puntos posteriores, la estructura de articulaciones que genera la aplicación no es la adecuada para poder implementar los scripts de cinemática inversa para el movimiento del avatar. Al intentar implementar los scripts de cinemática inversa, hubo problemas ya que la estructura de articulaciones que genera esta aplicación es incompleta, y no era posible asignar todos los scripts necesarios de la cinemática inversa a las articulaciones necesarias.

La segunda opción para el avatar, y la que se ha escogido finalmente para el proyecto, ha sido utilizar directamente la tienda de assets que ofrece Unity. En esta tienda, podemos encontrar tanto

contenido gratuito como de pago, de diferentes modelos, texturas, proyectos y material para utilizar en cualquier proyecto.

En este caso, se ha utilizado un modelo 3D de un avatar, que contiene la estructura correcta de articulaciones para poder aplicar los posteriores scripts de cinemática inversa.



**Figura 13 Avatar Unity store**

Para la correcta estructura de articulaciones del avatar, es necesario que este incluya las diferentes uniones entre extremidades de forma correcta. En la siguiente figura se muestra una representación de las diferentes uniones entre extremidades del avatar configurado para funcionar de manera correcta con la cinemática inversa.

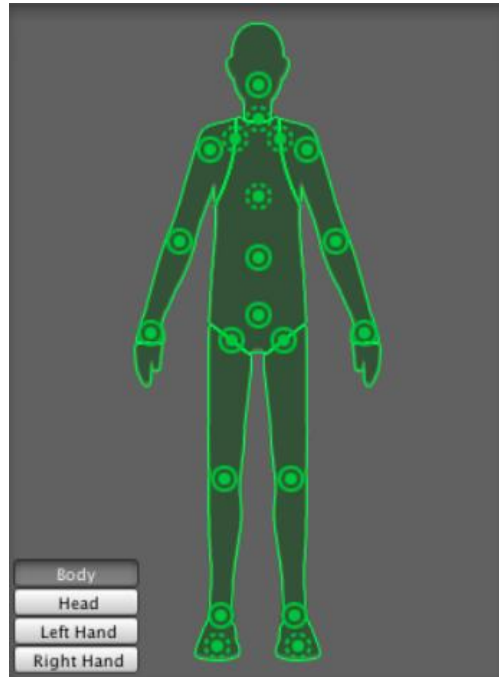


Figura 14. Estructura extremidades correcta avatar

Como se puede observar, es necesario que exista una unión de articulaciones en los codos y en las rodillas para que los scripts de cinemática inversa funcionen de manera correcta. Para el caso de generar el avatar mediante la aplicación *MakeHuman*, estas uniones entre articulaciones eran inexistentes, por lo que no se puede utilizar esta alternativa.

### 5.1.2 Implementación cinemática inversa

Dentro del desarrollo de aplicaciones en 3D, las animaciones de los personajes se consiguen en muchas ocasiones mediante la rotación de los ángulos de las articulaciones de un esqueleto a diferentes posiciones. La posición que adquiere una articulación hija cambia según a la rotación que sigue su padre, de esta manera, una cadena de articulaciones adquiere su posición final a partir de los ángulos y posiciones de cada una de las articulaciones individuales de las que se compone. Este método para posicionar un esqueleto de un personaje es conocido como *Forward Kinematics* o *Cinemática directa* [8].

No obstante, en muchas ocasiones, el proceso de fijar las posiciones de las articulaciones es útil realizarlo desde el punto de vista opuesto, es decir, dada una posición en un espacio, realizar el proceso hacia atrás para encontrar una forma de orientar las articulaciones para que la posición final acabe en esa posición. Esto puede ser útil, por ejemplo, cuando se desea que un personaje toque un objeto en un punto determinado por el usuario, o colocar su extremidad inferior en alguna superficie. Este proceso, es lo que se conoce como *Inverse Kinematics* o *Cinemática inversa* [9].

La cinemática inversa es una implementación importante del proyecto. Gracias a ella, será posible simular el movimiento de las articulaciones del avatar de la forma más realista posible, tratando de reflejar el movimiento que debe de realizar el paciente.

Para la implementación de la cinemática inversa en el proyecto, se han barajado diferentes scripts desarrollados para Unity que permitan implementarla de la manera más sencilla posible. Existen numerosas implementaciones, algunas de pago y otras gratuitas. Para el proyecto se ha optado por una implementación gratuita.

La implementación utilizada se denomina *YVTuber (Yume VTuber System)*, se trata de un proyecto Unity que aplica la cinemática inversa sobre un avatar, controlando su movimiento mediante los controladores de un visor de realidad virtual.



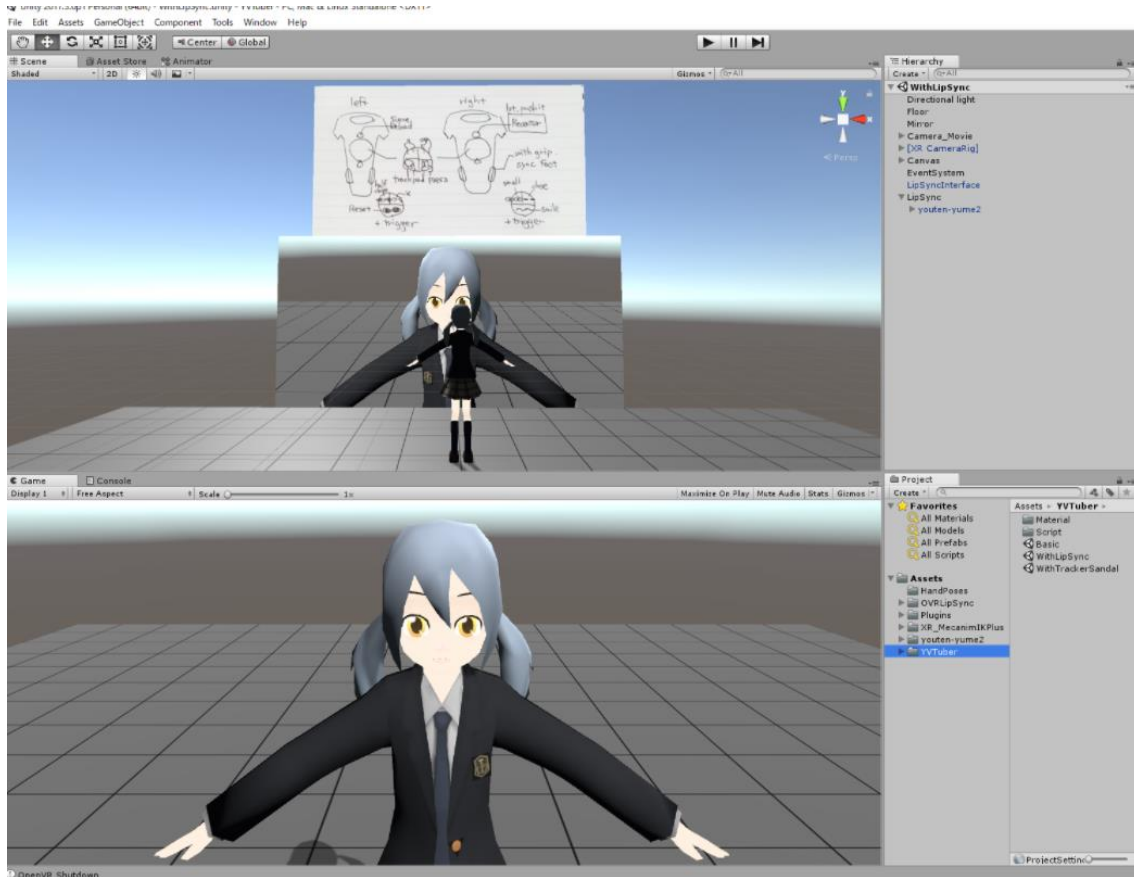


Figura 15. Proyecto Unity YVTuber

El proyecto puede ser descargado del portal de códigos *Github* de forma totalmente gratuita y se trata de un proyecto dirigido a *VTuber* (*Virtual Youtuber*). Un *youtuber* virtual es un *streamer* que se dedica a compartir contenido mediante el uso de modelos digitales generados por herramientas software. Se iniciaron en Japón, aunque actualmente se ha extendido y pueden encontrarse *VTubers* en la mayor parte del mundo.

La actividad principal de los *VTubers* incluye la comunicación con su audiencia de forma constante, pero también cantar, bailar o transmitir videojuegos.



Figura 16. Tsukino Mito, ejemplo de Youtuber Virtual

Mediante el uso de este proyecto se puede conseguir hacer uso de este avatar y realizar diferentes movimientos a través de un dispositivo de realidad virtual y sus controladores.

Para su implementación, es necesario disponer de un avatar con la estructura de articulaciones correcta, tal y como se comentó en el apartado anterior. Una vez se dispone del avatar con la estructura necesaria, se asignan los diferentes scripts de la cinemática inversa a cada una de las partes correspondientes del avatar. En el caso de este proyecto, en lugar controlar el movimiento del avatar mediante el uso de los controladores del visor de realidad virtual, el movimiento se va a controlar con el uso del teclado, por lo que ha sido necesario desarrollar unos scripts encargados de controlar este movimiento.

Estos scripts, se asignan a cada una de las extremidades del avatar y controlan la posición del avatar a lo largo de los ejes del espacio. A continuación, se muestra un ejemplo de uno de los scripts desarrollados:

```
public class ControlPd : MonoBehaviour
{
    public float velocidad = 0.02f;
    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey("q"))
        {
            Vector3 PosicionMano2 = new Vector3(0, 0, velocidad);

            transform.position -= PosicionMano2;
        }
    }
}
```

Figura 17. Script para controlar posición extremidades

En este script, se define una variable denominada *velocidad*, que permite controlar la velocidad de movimiento con la que se moverá la extremidad del avatar. En el método Update, que se ejecuta una vez por cada frame, se comprueba la entrada del teclado del usuario, y si se detecta la pulsación de la tecla “q”, se desplaza la parte del avatar asignada a lo largo del eje Z positivo.

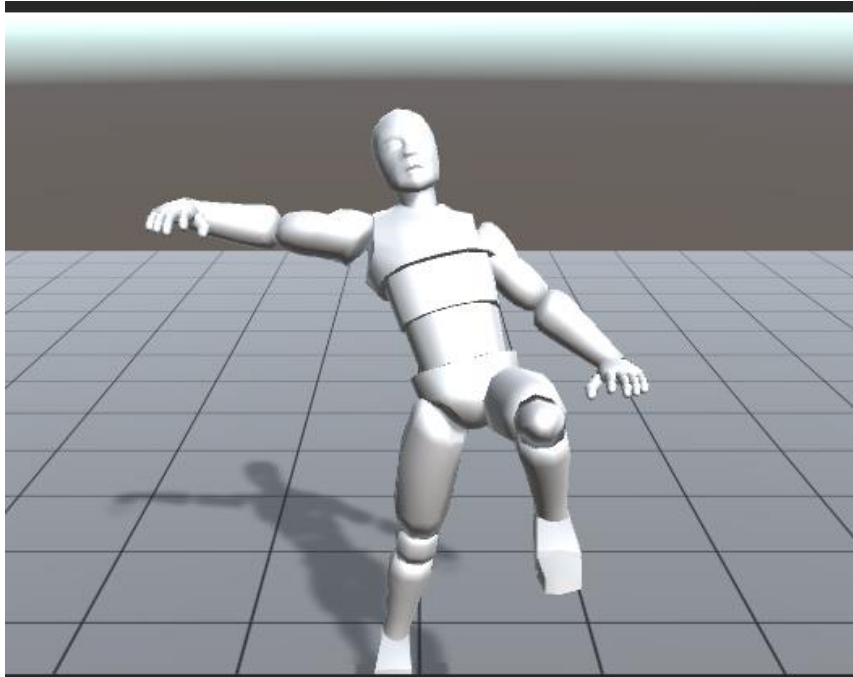


Figura 18. Cinemática inversa avatar

Para la implementación de la cinemática inversa, es necesario asignar los diferentes scripts del plugin utilizado, a una serie de *GameObjects*. El primero de ellos, será la cámara principal de la escena, al cual se debe de asignar el script *XR\_Basic\_Function\_CS*. Una vez asignado este script principal, debemos crear varios *GameObjects* hijos de la cámara de la escena y colocarlos a la altura necesaria del avatar para controlar cada una de las partes del cuerpo. Por tanto, se crea un componente para el cuerpo, 2 para los brazos, 2 para las piernas y un último componente que será el punto sobre el que pivota el avatar.

Estos scripts que se han asignado a los diferentes *GameObjects* creados anteriormente, son los responsables de los cálculos necesarios para aplicar la cinemática inversa, desplazando de la forma más realista posible todo el cuerpo, en función de la extremidad que se mueva en el avatar.

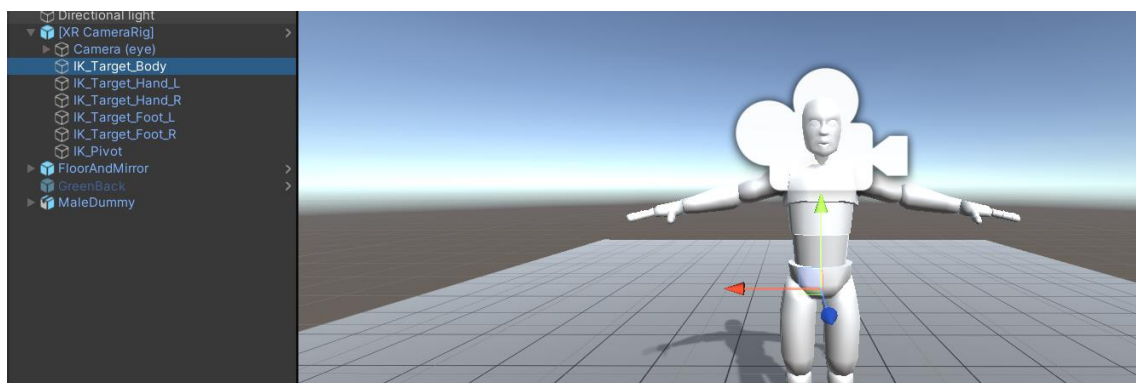


Figura 19. GameObjects para cinemática inversa

Finalmente, se asigna a un script final las diferentes partes del esqueleto del avatar para que pueda realizar el movimiento correspondiente de cada una de las partes de la cadena cinemática. Como se ha comentado anteriormente, el avatar debe de seguir una estructura del esqueleto válida para poder hacer funcionar el plugin.

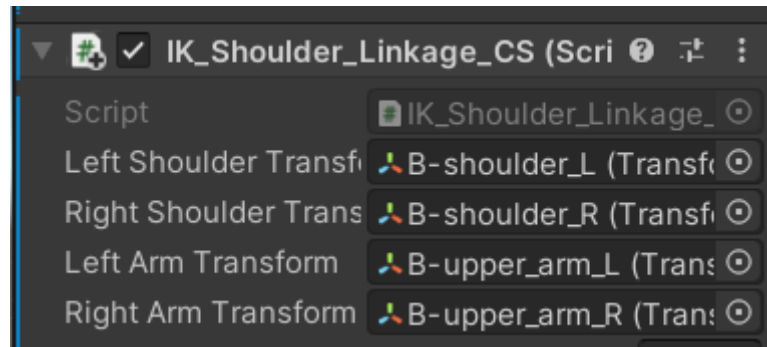


Figura 20. Asignación partes del esqueleto del avatar

### 5.1.3 Desarrollo matriz de esferas

Para la definición de los ejercicios, el avatar estará rodeado de una matriz de esferas, cada una de ellas seleccionable por el usuario para definir el ejercicio. Las esferas tendrán una opacidad del 20% para que sean perceptibles por el usuario, pero no molesten a la hora de definir un ejercicio.

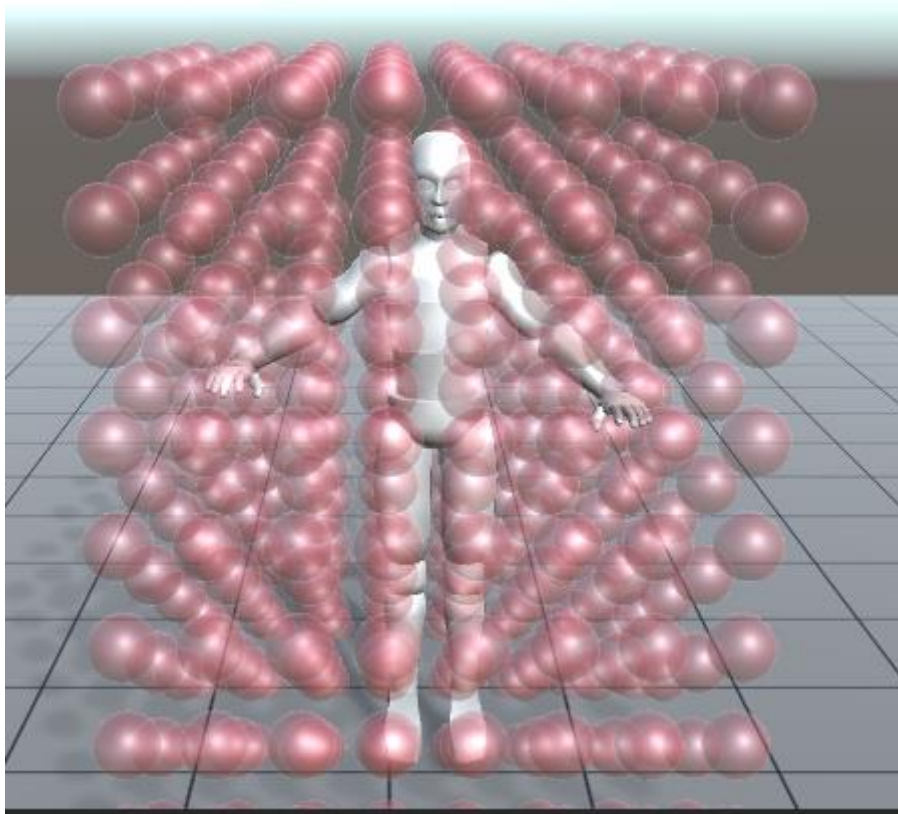
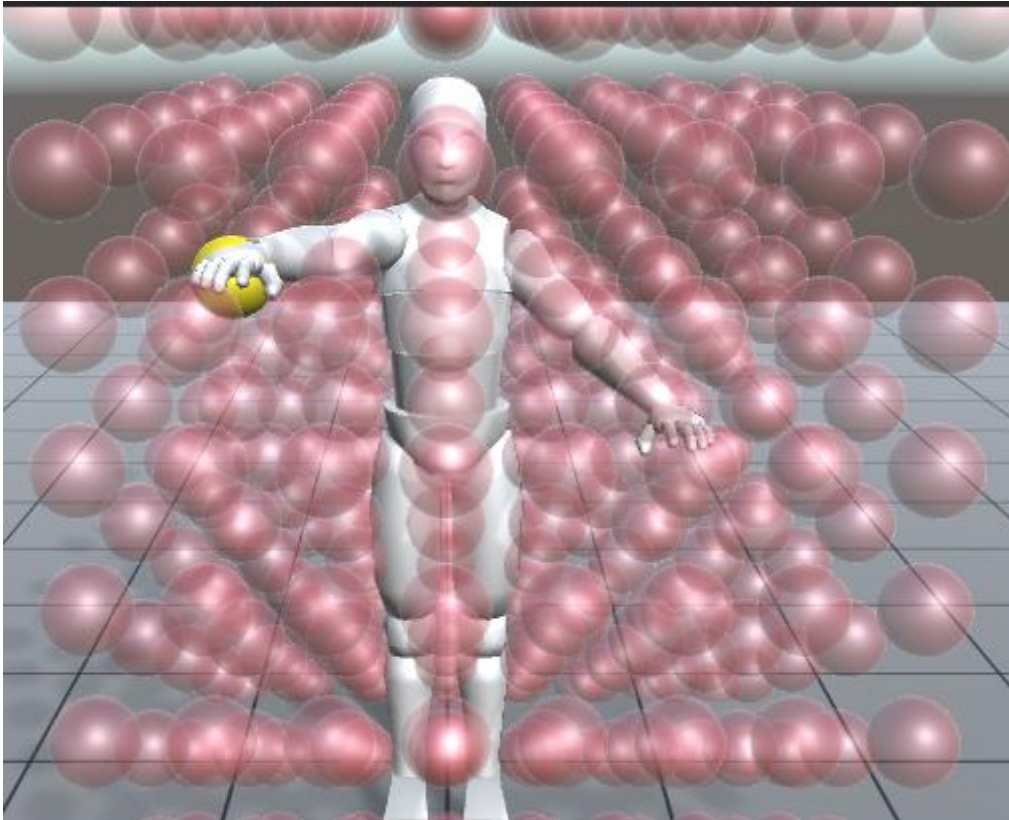


Figura 21. Matriz de esferas

Las esferas se encuentran equidistantes entre sí, e implementan un sistema de colisiones mediante el cual se detecta si alguna extremidad del avatar ha entrado en contacto con una esfera, si ocurre esta situación, la esfera correspondiente se ilumina de color amarillo para advertir al usuario de la esfera que está siendo seleccionada por el avatar.



**Figura 22. Avatar seleccionando esfera**

De esta manera, el fisioterapeuta controlará al avatar como si se tratara del paciente, seleccionando las esferas que crea convenientes para el ejercicio que se está definiendo.

Para que la aplicación sea compatible con el proyecto anterior, la matriz de esferas debe de tener el mismo tamaño, por ello, se han utilizado los mismos cálculos para generarla, teniendo en cuenta la distancia entre extremidades, la altura del avatar y separando de forma equidistante cada esfera. En concreto, tenemos una matriz de esfera con dimensión  $9 \times 8 \times 5$ , lo que nos da un total de 360 posiciones donde se puede encontrar la esfera seleccionada. A cada esfera se le asigna un índice o número de esfera, que permite mediante el sistema de colisiones poder almacenar la esfera seleccionada por el avatar. Una vez la esfera es seleccionada y almacenada mediante el uso de la tecla adecuada, se almacena ese índice en un array, el cual será utilizado posteriormente en el script de creación del fichero JSON para generar la sesión de entrenamiento con las diferentes esferas que se han seleccionado.

El fichero JSON será creado en el directorio por defecto, que a la vez es el mismo que utiliza la aplicación de validación para de esta manera, leer directamente el último fichero generado por la aplicación de creación de ejercicios de forma automática. Para cargar la sesión de entrenamiento en la aplicación del cliente, será necesario transportar este fichero al directorio de la aplicación del cliente. Como se trata de un fichero JSON de tamaño reducido, es sencillo de transportar directamente por la red mediante correo electrónico por ejemplo.

#### **5.1.4 Implementación sistema de colisiones**

Una vez generada la matriz de esferas, es necesario desarrollar un sistema para controlar la colisión de las extremidades del avatar con la esfera que se desea seleccionar para definir el ejercicio. Para desarrollar este sistema, se ha hecho uso del concepto de *Colliders* en Unity, que permite controlar las colisiones en videojuegos 3D.

Cuando se desarrolla un videojuego, una de las tareas más comunes a desarrollar es detectar las colisiones que se producen entre los diferentes elementos que componen la escena, esto

proporciona la lógica de interacción existente entre el jugador y los elementos que componen el entorno de la escena.

Los *Colliders* son áreas que cubren al *GameObject* de Unity, y se utilizan para poder detectar las colisiones que se producen entre los diferentes objetos del juego. Un *Collider* se compone en primer lugar de una forma, donde lo recomendable es que esta forma sea lo más parecida al objeto que recubre y, por otro lado, se compone de un material físico determinado, que básicamente proporciona las diferentes características de fricción o rebote.

A continuación, se presentan los principales *Colliders* que encontramos en Unity:

**Box collider:** *collider* con forma de caja o cubo alrededor del objeto

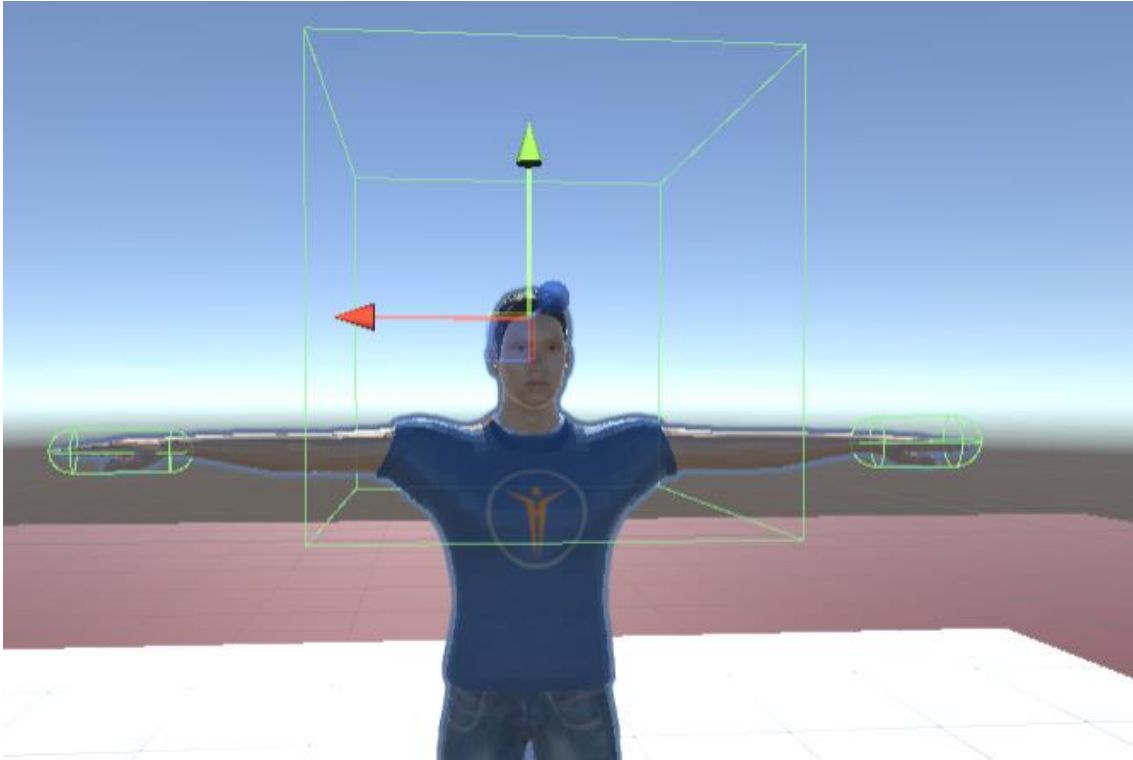


Figura 23. Box collider

**Sphere collider:** forma esférica o de pelota

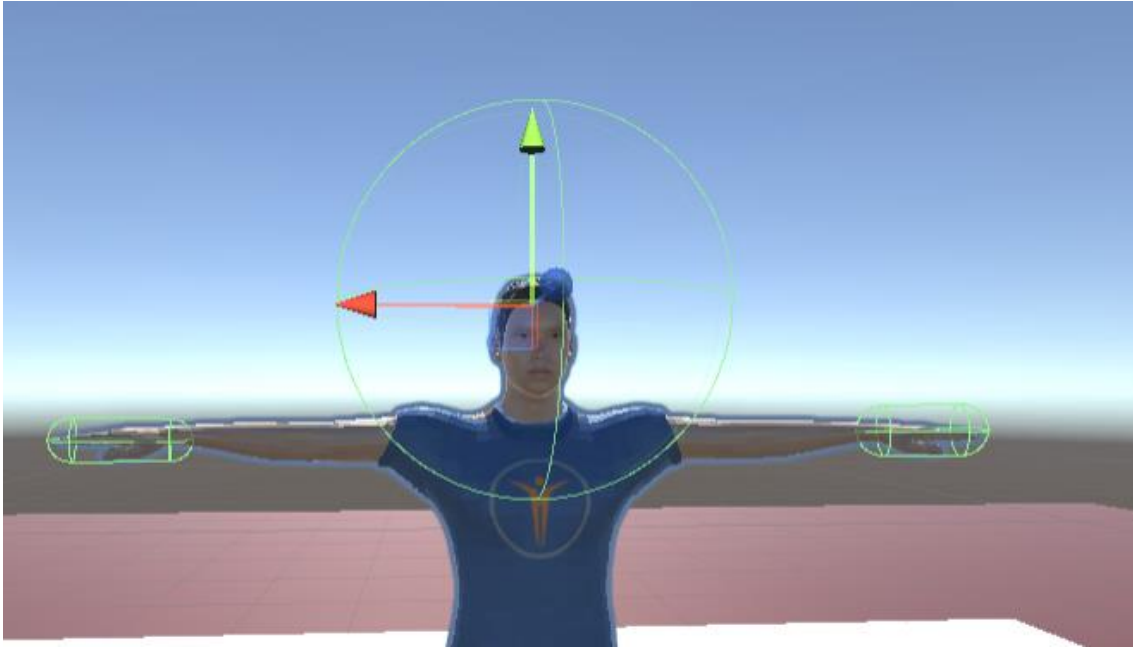


Figura 24. Sphere Collider

**Capsule collider:** tiene forma de cápsula o elipsoide

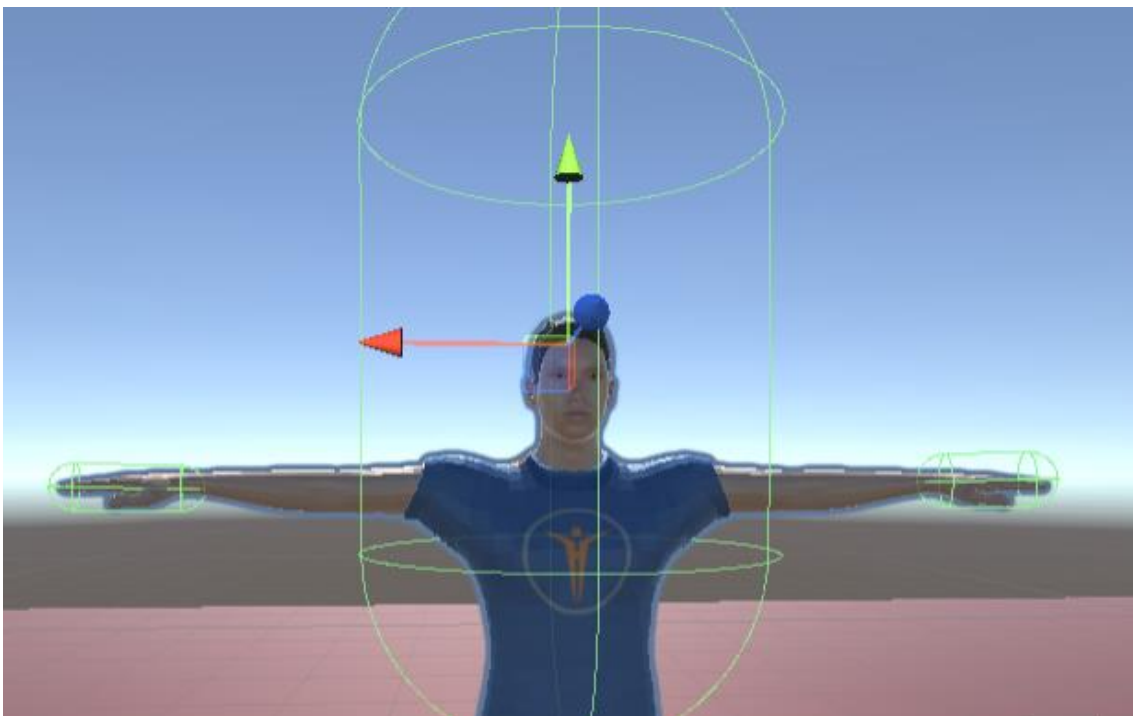


Figura 25. Capsule collider

**Mesh collider:** este colisionador de malla intenta acoplarse a la forma que tiene el objeto al que se le asigna, mediante este tipo de colisionador se pueden obtener colisiones de manera muy precisa, sin embargo, consume muchos recursos de computación

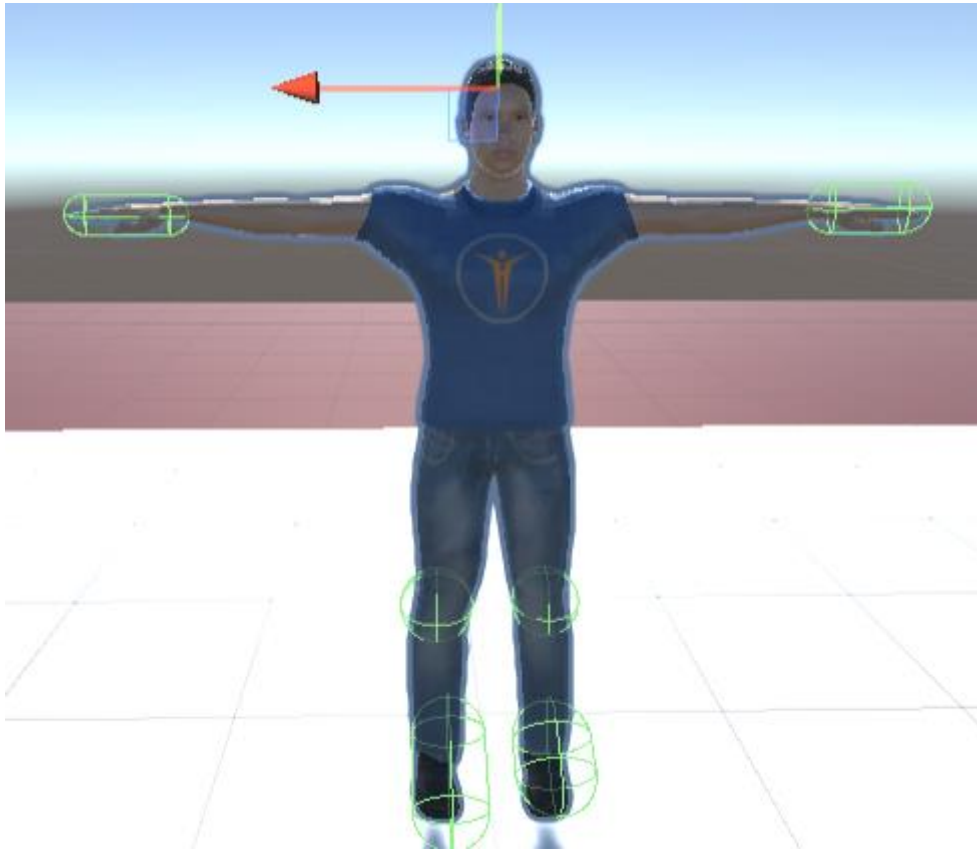


Figura 26. Mesh collider

Aparte de los elementos *Colliders*, es necesario que alguno de los objetos que van a formar parte de las colisiones, haga uso del componente *Rigidbody*. Este componente básicamente permite añadir al *GameObject* algunas propiedades que el objeto tendrá en la realidad, como puede ser por ejemplo gravedad, fricción o masa.

Para el proyecto, en el script de creado de la matriz de esferas, cada vez que se crea una esfera se le asigna un *Sphere Collider* del mismo radio que la esfera y un *Rigidbody* con las diferentes características físicas necesarias para poder detectar la colisión con alguna de las extremidades del avatar.

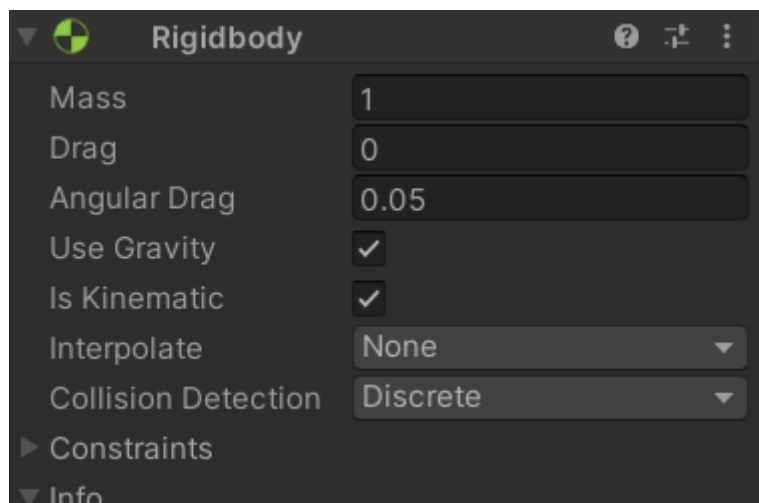


Figura 27. Rigidbody esfera

Para el *Rigidbody*, se activa la opción de *Is Kinematic*, de esta manera las esferas de la matriz no se verán afectadas por el motor de físicas, es decir, los parámetros de masa, gravedad o fricción



no se tendrán en cuenta, sin embargo, el componente *Rigidbody* nos permitirá detectar las colisiones con los demás objetos.

También se ha añadido un *Box collider* en cada una de las extremidades del avatar y se intenta ajustar lo máximo posible a su forma. No se ha utilizado un *Mesh collider*, ya que como se comentó anteriormente, este tipo de componentes consumen muchos recursos, y para detectar una colisión simple de una esfera con una extremidad, no es necesario un colisionador muy complejo.

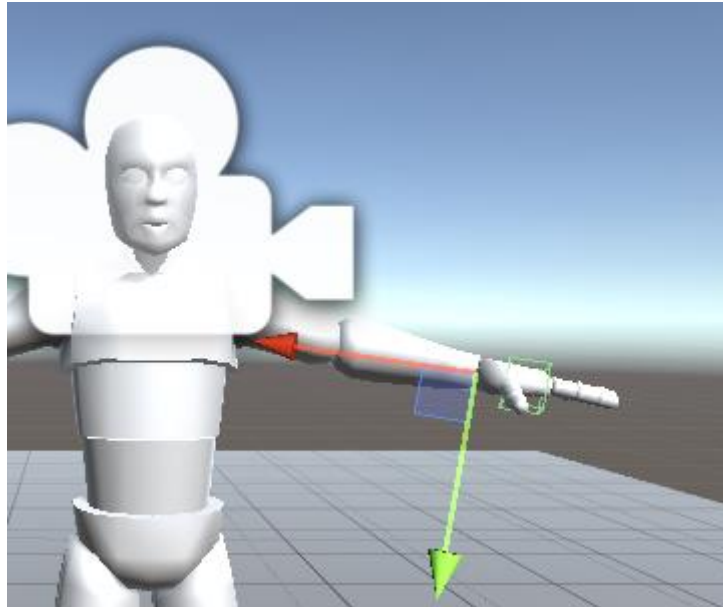


Figura 28. Box collider extremidad derecha

Una vez asignados los diferentes componentes *Collider* y *Rigidbody*, mediante el uso de los métodos *OnTriggerEnter()* y *OnTriggerExit()* se puede controlar las colisiones que se producen. Este método se dispara, cuando un collider de un objeto entra en el área de otro. De esta manera se puede controlar que el collider de las extremidades, entre en el área de alguna de las esferas y por tanto detectar la esfera que está tocando el avatar.

### 5.1.5 Implementación sistema definición de ejercicios

Para la definición de ejercicios, el objetivo es que sea lo más sencillo posible y cómodo para el usuario. Para ello, el fisioterapeuta controlará al avatar para seleccionar la esfera deseada, y una vez fijada, pulsará un botón del teclado para definir esa esfera como parte del ejercicio. Si el usuario desea definir más esferas para el ejercicio, debe de repetir el proceso. Si en caso contrario quiere definir un nuevo ejercicio o terminar la sesión de entrenamiento, deberá de usar la tecla asignada para ello.

Para la detección de la esfera seleccionada por el avatar, se realiza un sistema de colisiones, de tal manera que se almacena la última esfera con la que ha colisionado el avatar. Cuando el usuario pulsa la tecla para guardar la esfera seleccionada, se guarda la posición de esa esfera en un array, para posteriormente escribirlo en el fichero JSON de la sesión de entrenamiento.

A continuación, se presenta un diagrama de flujo para explicar el principio de funcionamiento de la aplicación.

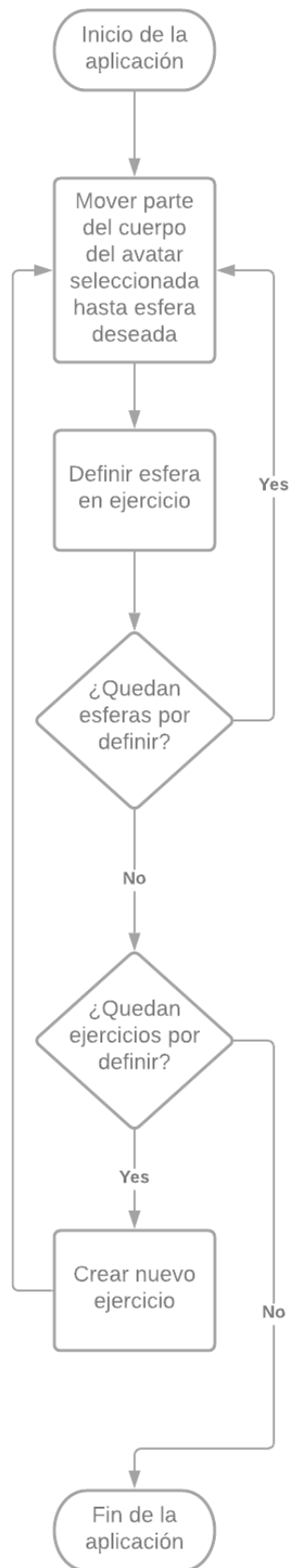
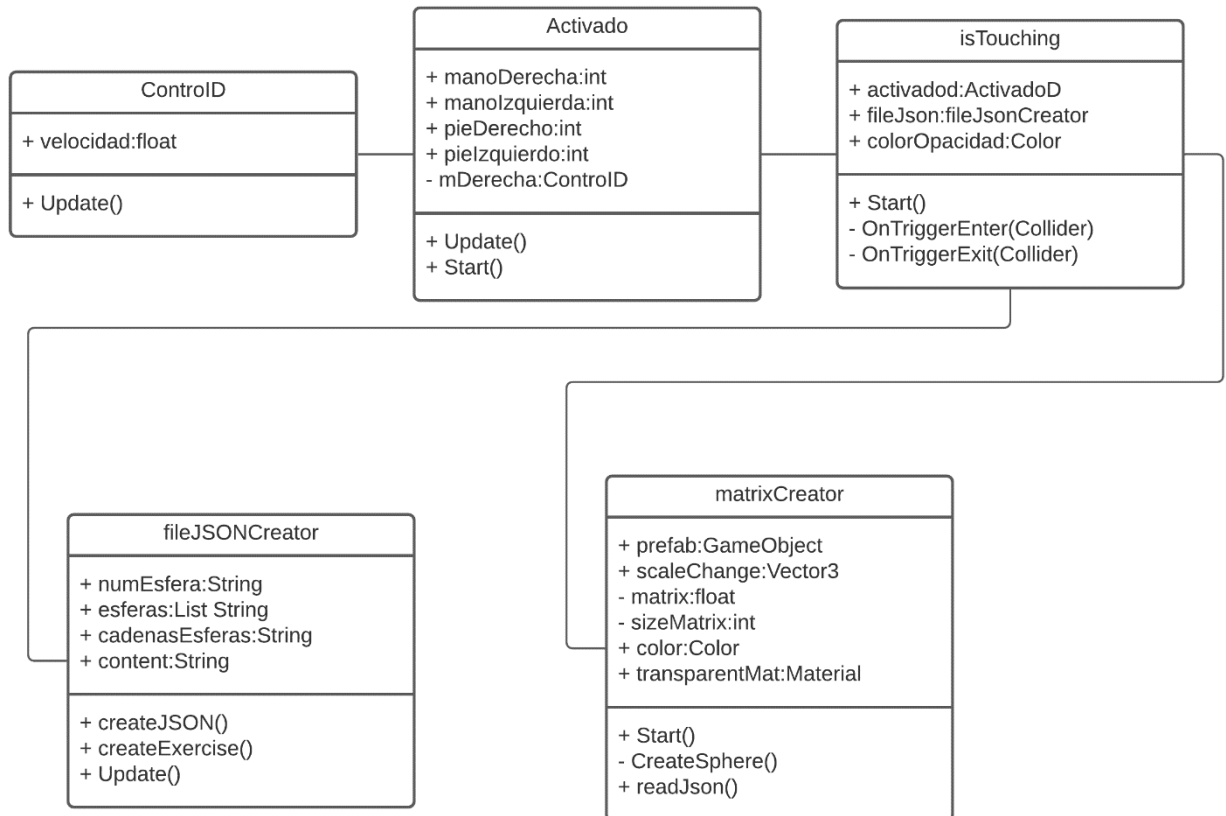


Figura 29. Diagrama de flujo aplicación ejercicios

Para detallar el funcionamiento de la aplicación desarrollada, se muestra el diagrama de clases del software:



**Figura 30. Diagrama de clases módulo ejercicios**

La aplicación para definición de ejercicios se compone de 5 clases fundamentales, a continuación, se muestra una tabla explicativa de cada una de ellas.

Clase <b>matrixCreator</b>	
Función	Descripción
Start()	Inicializa las variables globales y calcula el número de elementos de la matriz de esferas según la distancia de las extremidades del avatar.
CreateSphere()	Crea una esfera en cada una de las posiciones de la matriz calculada anteriormente. Añade además, los componentes y parámetros necesarios a cada esfera para poder controlar las colisiones con el avatar.

**Tabla 1. Clase matrixCreator**

Este script, se asocia al GameObject del avatar y se inicia al arrancar la aplicación.

<b>Clase isTouching</b>	
Start()	Inicializa las variables
OnTriggerEnter()	Este método se ejecuta cuando se detecta una colisión de la esfera con el avatar, en caso de ejecutarse cambia el color de la esfera correspondiente en amarillo y almacena el índice de esta.
OnTriggerExit()	Este método se ejecuta en caso de detectar que se está saliendo de la colisión con el avatar, en este caso se vuelve a cambiar el color de la esfera a rojo y la opacidad al 20%

**Tabla 2. Clase isTouching**

La clase *isTouching*, se asocia a cada esfera individualmente en el método *CreateSphere* de la clase *matrixCreator*, con esta clase se controla las colisiones existentes entre las extremidades del avatar y las esferas de la matriz, de tal manera que se puede almacenar la esfera seleccionada por el avatar para poder escribirla en el fichero JSON de salida.

<b>Clase fileJsonCreator</b>	
createJSON()	Método que define la ruta del fichero JSON de salida, además concatena al string final los caracteres necesarios y escribe el fichero en la ruta indicada
createExercise()	En caso de que el usuario cree un nuevo ejercicio, se ejecuta este método, que incluye el contenido necesario al fichero JSON para crear un ejercicio
Update()	Se ejecuta una vez por frame, permite controlar las teclas pulsadas por el usuario, en caso de que se pulse la tecla “espacio”, se añade la esfera seleccionada a la sesión, en caso de pulsar “shift”, se crea un nuevo ejercicio, en caso de pulsar “enter” se termina la sesión y se crea el fichero JSON de salida

**Tabla 3. Clase fileJsonCreator**

La clase *fileJsonCreator* permite el control de la definición de ejercicios, de tal manera que en función del comportamiento del usuario se realiza una función. En el caso que el usuario pulse la tecla “espacio”, se añade a la sesión de entrenamiento la última esfera seleccionada por el avatar, y se concatena al string final que se escribirá en el fichero de salida. En el caso de que el usuario cree un nuevo ejercicio, se completa el string para añadir este ejercicio. Y finalmente, si el usuario decide terminar la sesión, se ajusta el string final y se ejecuta la función para escribir el fichero de salida en la ruta indicada.

<b>Clase Activado</b>	
Update()	Este método se utiliza para controlar que extremidad que ha seleccionado para realizar el movimiento

Start()	Se ejecuta al comienzo para inicializar las variables a los valores necesarios
---------	--

La clase *Activado*, permite realizar el control de la extremidad que ha sido seleccionada para el usuario para posteriormente realizar el movimiento deseado.

Clase Contro	
Update()	Se utiliza para controlar la dirección en la que el usuario deseado mover la extremidad seleccionada. Según la tecla presionada, se mueve la extremidad en una dirección

Mediante la clase *Contro*, se detecta la tecla apretada por el usuario y se desplaza la extremidad seleccionada previamente en la dirección deseada.

### 5.1.6 Desarrollo escritura archivo JSON

En la aplicación del paciente, se utiliza un archivo JSON para leer la estructura de la sesión de entrenamientos. Para que la aplicación de este proyecto sea compatible con la anterior, y se pueda leer los ficheros generados, es necesario que la estructura sea la misma.

En la siguiente figura, se muestra un ejemplo del archivo JSON necesario para la aplicación del cliente.

```
Ejercicio.json
1 [{"motor_rehabilitation":[{
2
3
4     "general_description": "This session will have two exercises! One for arms and one for legs!",
5     "number_of_exercises": 2,
6     "exercises": [{
7         "exercise_description": "Exercise 1",
8         "sphere_position": [213, 253, 257, 258, 266],
9         "repetitions": 3,
10        "user_position": "standing",
11        "pause": 10
12    },
13    {
14        "exercise_description": "Exercise 2",
15        "sphere_position": [213, 281, 254],
16        "repetitions": 3,
17        "user_position": "standing",
18        "pause": 10
19    }
20    ]
21 }],
```

Figura 31. Ejemplo archivo JSON

La estructura del archivo se compone básicamente de un campo para la descripción de la sesión, y otro campo para definir el número de ejercicios.

Dentro de los ejercicios, encontramos campos para la descripción del ejercicio, un array con las posiciones de las esferas que están definidas para ese ejercicio, el número de repeticiones del ejercicio, la posición del usuario, y un tiempo de pausa que hay entre el final de un ejercicio y el comienzo del siguiente.

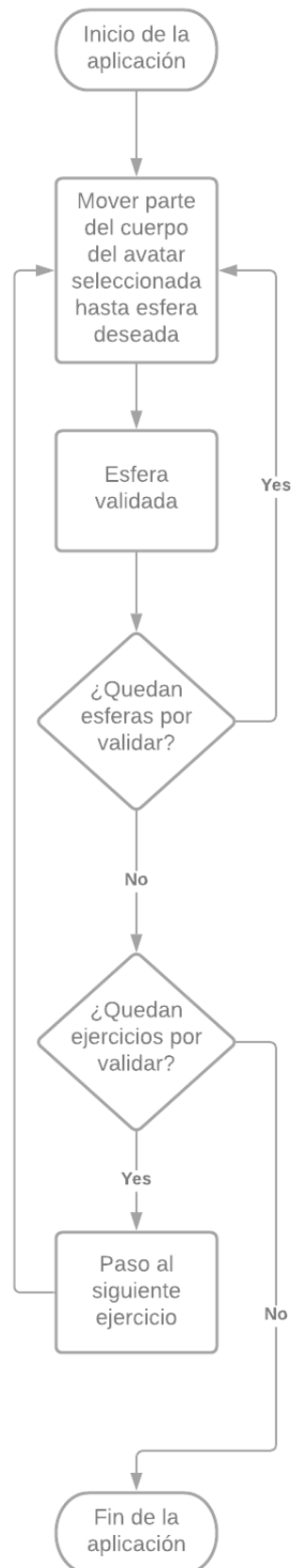
Para la escritura del archivo JSON en la aplicación de este proyecto, se almacenan las posiciones de las esferas para cada ejercicio en un array, una vez el usuario decide terminar la sesión y generar el fichero, simplemente aprieta una tecla y se escribe el fichero JSON en el directorio por defecto. Este proceso se realiza mediante la concatenación de cadenas de texto según el comportamiento del usuario para formar la estructura correcta del archivo de salida.



### **5.1.7 Desarrollo aplicación validación de ejercicios**

Además del módulo de definición de ejercicios por parte del fisioterapeuta, se ha desarrollado otro módulo que permite la validación de los ejercicios definidos. Esta aplicación utiliza el fichero JSON generado por la aplicación anterior que almacena las posiciones de las esferas definidas para todos los ejercicios. Una vez leído el fichero, genera la matriz de esferas y comienza a marcar de un color llamativo las esferas definidas en el archivo. El usuario debe realizar el proceso inverso a la definición de ejercicios, es decir, debe de utilizar el avatar para realizar el ejercicio como si se tratase del paciente, y de esta manera comprobar que todas las esferas deseadas para el ejercicio propuesto se han marcado correctamente y son realizables por parte del paciente.

Al igual que en el caso anterior, se presenta un diagrama de flujo para mostrar la lógica que seguirá la aplicación.



**Figura 32. Diagrama de flujo aplicación de validación**

Para el desarrollo de la aplicación de validación de ejercicios, se muestra el diagrama de clases en la siguiente figura:

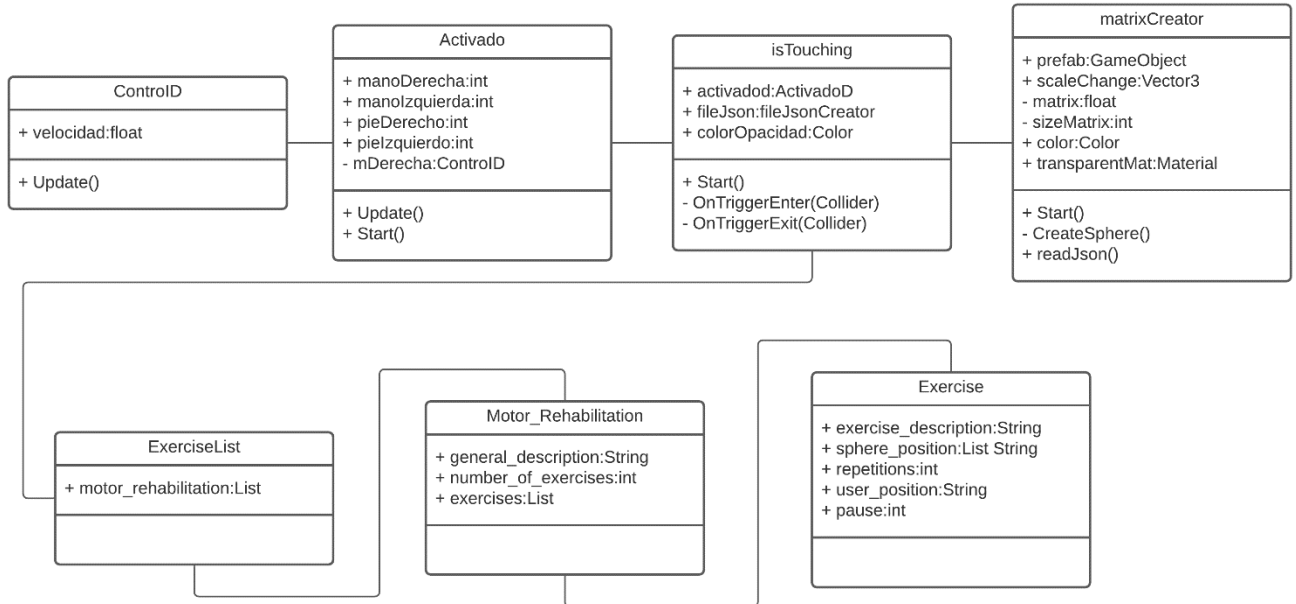


Figura 33. Diagrama de clases aplicación de validación

A continuación, se muestra una tabla explicativa con cada una de las clases de la aplicación de validación.

Clase matrixCreator	
Función	Descripción
Start()	Inicializa las variables globales y calcula el número de elementos de la matriz de esferas según la distancia de las extremidades del avatar.
CreateSphere()	Crea una esfera en cada una de las posiciones de la matriz calculada anteriormente. Añade además, los componentes y parámetros necesarios a cada esfera para poder controlar las colisiones con el avatar.
readJson()	Lee el fichero JSON en la ruta indicada
Update()	Se ejecuta una vez por frame, y actualiza el valor de la esfera actual, en caso de que queden esferas por leer en el fichero.

Tabla 4. Clase matrixCreator aplicación validación

Se utiliza la misma clase *matrixCreator* que en el caso de la aplicación de definición de ejercicios, ya que la matriz de esferas debe de ser exactamente la misma para poder llevar a cabo las validaciones necesarias, sin embargo, se añaden nuevos métodos, *readJson*, que permite leer el contenido del fichero JSON con las esferas y ejercicios de la correspondiente sesión de



entrenamiento, y el método *Update*, que permite actualizar el valor de la última esfera leída del fichero.

<b>Clase ExerciseList</b>	
<a href="#">Clase Motor_Rehabilitation</a>	Permite la creación de arraylist para almacenar la información de los ejercicios del fichero JSON
<a href="#">Clase Exercise</a>	Crea un arraylist para almacenar la información sobre las esferas de la sesión de entrenamiento

**Tabla 5. Clase ExerciseList**

Mediante la clase *ExerciseList* se crean las estructuras de arraylist necesarias para poder leer de manera correcta el fichero JSON y almacenar la información de este.

<b>Clase isTouching</b>	
Start()	Inicializa las variables
Update()	Se ejecuta 1 vez por frame, y cambia el color de la esfera de la matriz de esferas a amarillo, según la esfera leída del fichero JSON
OnTriggerEnter()	Este método se ejecuta cuando se detecta una colisión de la esfera leída del fichero JSON con el avatar, en el caso de que se produzca la colisión, se actualiza el valor de la esfera actual con la siguiente esfera del fichero JSON, en caso de que no queden más esferas se pasa a leer las esferas del siguiente ejercicio.
OnTriggerExit()	Este método se ejecuta en caso de detectar que se está saliendo de la colisión con el avatar, en este caso se vuelve a cambiar el color de la esfera a rojo y la opacidad al 20%

**Tabla 6. Clase isTouching aplicación validación**

La clase *isTouching* de la aplicación de validación de ejercicios, permite controlar todas las esferas que se han almacenado en el fichero JSON, de tal manera que se resalta de color amarillo la última esfera leída del fichero, en caso de que el avatar colisione con alguna extremidad con esta esfera, se actualiza el valor de la esfera actual con la siguiente esfera del fichero JSON y se repite el mismo proceso. En caso de que no existan más esferas por leer en el ejercicio actual, se pasa al siguiente ejercicio y se repite el mismo proceso. Por último, si no quedan ejercicios ni esferas por leer del fichero, se finaliza la aplicación.

<b>Clase Activado</b>	
Update()	Este método se utiliza para controlar que extremidad que ha seleccionado para realizar el movimiento



Start()	Se ejecuta al comienzo para inicializar las variables a los valores necesarios
---------	--

**Tabla 7. Clase Activado aplicación validación**

Al igual que en el caso del módulo de la aplicación de definición de ejercicios, se usa la misma clase *Activado* que permite el control de la extremidad seleccionada por el usuario.

<b>Clase Contro</b>	
Update()	Se utiliza para controlar la dirección en la que el usuario deseado mover la extremidad seleccionada. Según la tecla presionada, se mueve la extremidad en una dirección

**Tabla 8. Clase Contro aplicación validación**

Para el control del movimiento de las extremidades, también se hace uso de la misma clase *Contro* que en el caso de la aplicación de definición de ejercicios.

## 5.2 Resultados aplicación para definición de ejercicios

La aplicación final recoge las posiciones de las esferas y los ejercicios definidos por el fisioterapeuta y escribe el resultado en un fichero JSON para que pueda ser leído por el módulo de validación o por la aplicación de realidad virtual del cliente.

El uso de la aplicación es sencillo, y solamente necesita un ordenador con teclado y ratón para ejecutarse, facilitando al terapeuta la definición de sesiones de ejercicios.

Cuando el usuario inicia la aplicación de definición de ejercicios, se genera la matriz de esferas y el avatar, en este momento, el usuario puede seleccionar la extremidad que desea mover mediante el teclado.

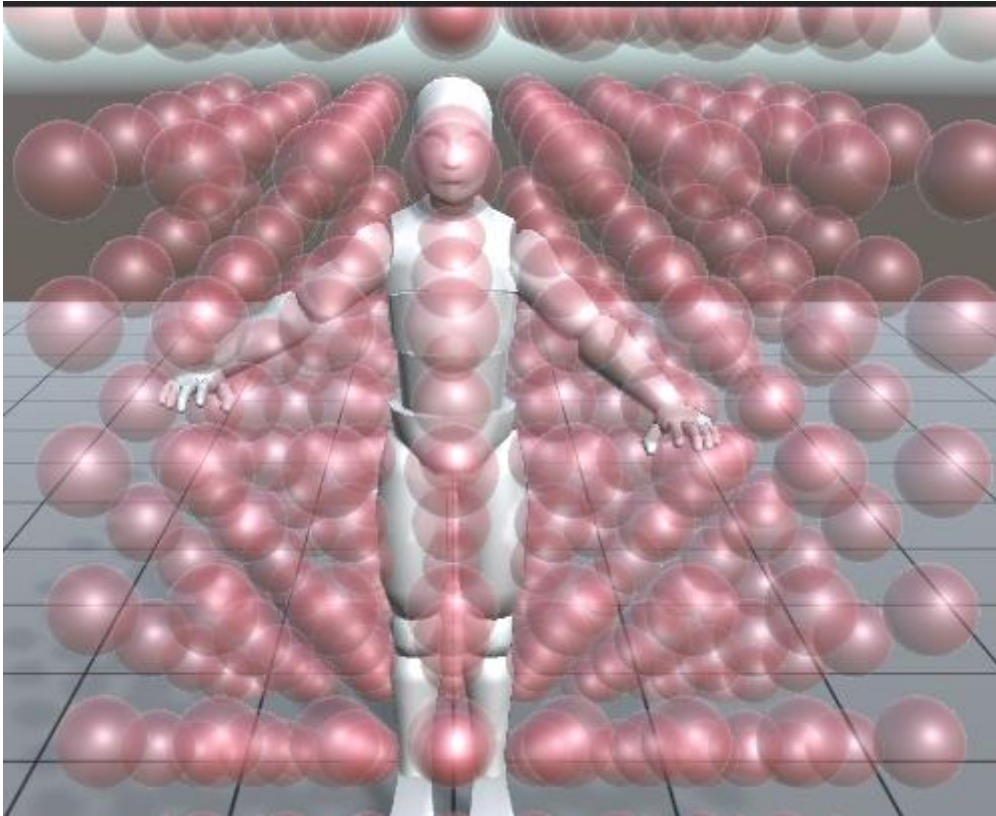


Figura 34. Arranque aplicación definición de ejercicios

A continuación, se muestran las teclas asignadas para el control de la aplicación.



Figura 35. Teclas control aplicación

Las teclas marcadas con color rojo permiten al usuario seleccionar la extremidad que desea desplazar:

- Z: pierna izquierda
- X: pierna derecha
- C: brazo izquierdo
- V: brazo derecho

Una vez seleccionada la extremidad del avatar a mover, se usan las teclas marcadas de color verde para desplazarla en el espacio.

Cuando se ha movido la extremidad hasta la esfera deseada, se utilizan las teclas amarillas para el control del ejercicio:

- Espacio: definir esfera actual dentro del ejercicio
- Shift: comenzar un nuevo ejercicio
- Enter: terminar la sesión y escribir el fichero JSON

Una vez se ha terminado la sesión, se genera en el directorio de la aplicación el fichero JSON correspondiente de la siguiente forma:

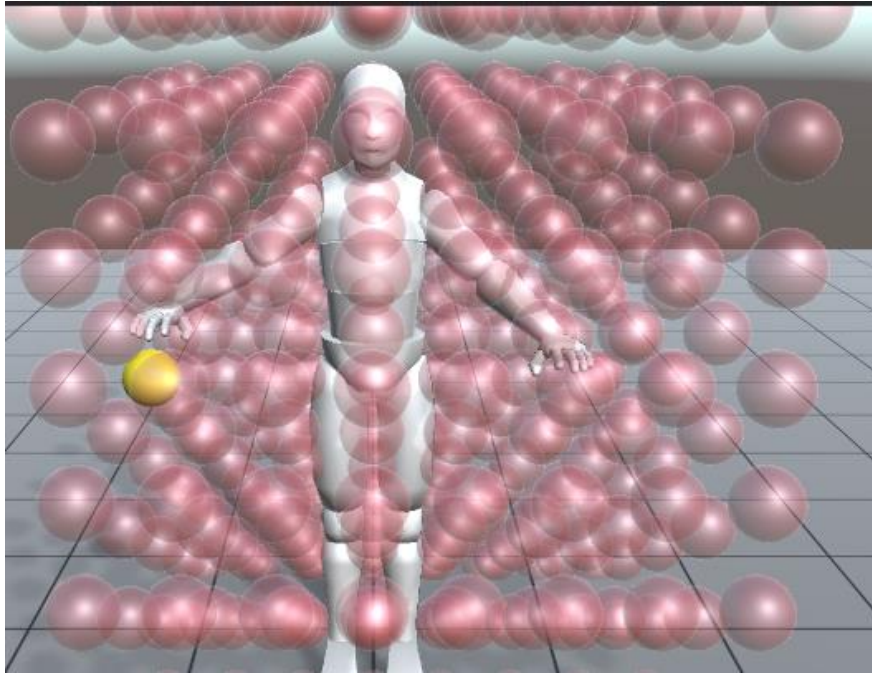
```
"motor_rehabilitation": [  
  {  
    "general_description": "This session will have one exercise!",  
    "number_of_exercises": 1,  
    "exercises": [  
      {  
        "exercise_description": "Exercise 1",  
        "sphere_position": [  
          337,  
          342,  
          352  
        ],  
        "repetitions": 3,  
        "user_position": "standing",  
        "pause": 10  
      }  
    ]  
  }  
]
```

Figura 36. Fichero JSON generado por aplicación

Para los campos de descripción, posición del usuario y pausa entre ejercicios, se generan actualmente de manera estática, no es posible modificar estos valores, se propone como futuras mejoras para el proyecto.

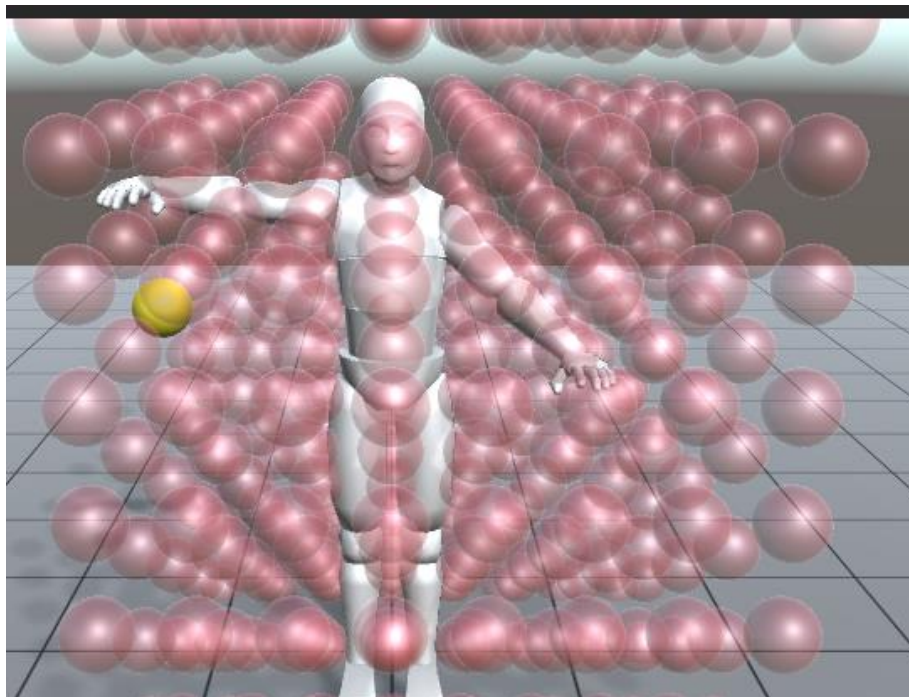
### 5.3 Resultados aplicación para validación de ejercicios

Para el caso del módulo de validación de ejercicios, el funcionamiento de la aplicación es bastante similar. Al arrancar la aplicación, se genera la misma matriz de esferas y el avatar, sin embargo, en este caso se lee el fichero JSON automáticamente generado por el módulo anterior y se marca la primera esfera de color amarillo leída por el fichero.



**Figura 37. Aplicación validación de ejercicios**

Una vez se ha marcado la primera esfera, el usuario debe de utilizar los mismos controles que en el caso del módulo anterior para tocar la esfera marcada, cuando se produce la colisión, se lee la siguiente esfera del fichero JSON y se marca de color amarillo en la matriz de esferas.



**Figura 38. Siguinte esfera a validar**

Si se han leído todas las esferas marcadas para un ejercicio, la aplicación pasa automáticamente al siguiente ejercicio definido en el archivo, y marca de nuevo la esfera correspondiente de color amarillo. En caso de que no existan más esferas o ejercicios por leer, la aplicación concluye y se da por terminada la validación. En este punto, el usuario ha comprobado que se hayan marcado correctamente todas las esferas necesarias para el ejercicio, además, comprueba que los movimientos definidos son los correctos para el paciente y son realizables.

## Capítulo 6. Conclusiones y propuesta de trabajo futuro

Tras la dura pandemia vivida durante los últimos meses, la actividad de rehabilitación o ejercitación muscular de algunos grupos sociales se ha visto altamente afectada.

En este proyecto se ha continuado con el trabajo de un proyecto anterior, donde se proponía el desarrollo de una aplicación de realidad virtual para realizar ejercicios de rehabilitación de pacientes con problemas físicos. Para este proyecto, se ha desarrollado la parte del fisioterapeuta, mediante la cual se hace posible la definición de los diferentes ejercicios que el paciente podrá realizar de forma remota. Además, se ha desarrollado un módulo secundario para poder validar estos ejercicios y comprobar de alguna manera que se han definido correctamente y son realizables por el paciente.

El desarrollo de la aplicación ha sido posible mediante el uso del motor gráfico Unity, y haciendo usas de técnicas de animación de personajes en 3D como la cinemática inversa, que ha permitido simular el movimiento del avatar de una forma realista para asimilarlo al movimiento del paciente.

Tras el estudio de las diferentes opciones para el desarrollo de la aplicación, se ha optado por hacer uso de periféricos básicos como teclado y ratón para que el fisioterapeuta pueda definir los ejercicios sin necesidad de adquirir un nuevo hardware. Sin embargo, la aplicación se ha desarrollado para que sea fácilmente adaptable a otros medios de tracking o movimiento del avatar que se proponen para un trabajo futuro.

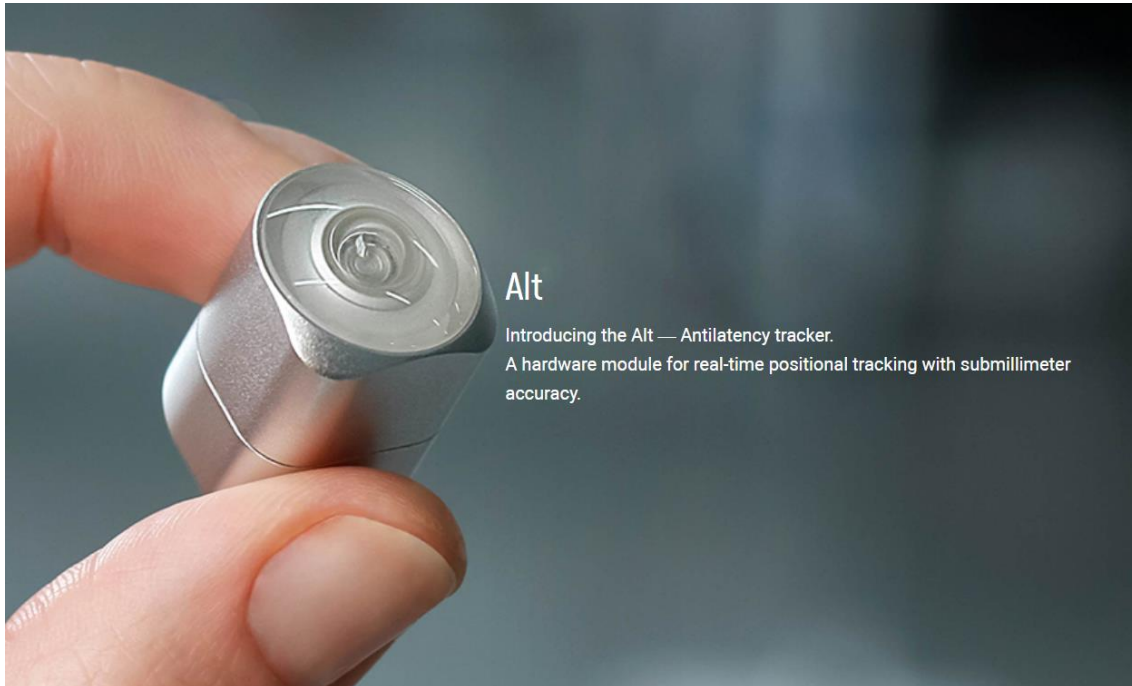
El resultado del proyecto ha sido una aplicación fácil de usar, y que además es de bajo coste ya que no necesita de un hardware muy potente para ser ejecutada. Aunque la aplicación ha sido desarrollada para ejecutarse en un ordenador, se propondrá como trabajo futuro lanzar la aplicación para diferentes dispositivos como smartphones o dispositivos de realidad virtual. Esta aplicación supondrá una gran ventaja para los fisioterapeutas, que podrán realizar de manera remota la definición de las sesiones de entrenamiento para sus pacientes, proporcionando además de un ahorro en tiempo y en desplazamiento, una forma mucho más óptima de realizar las sesiones de los pacientes y el seguimiento de estas. De esta manera, tanto los pacientes como el fisioterapeuta, reducirán los desplazamientos hasta las clínicas, reduciendo el contacto social para evitar contagios durante la pandemia y ahorrando costes y tiempos en desplazamientos.

El uso de una tecnología como la realidad virtual para estos proyectos, proporciona a los pacientes una motivación extra a la hora de realizar los ejercicios y por tanto unos mejores resultados. Por parte de los fisioterapeutas, supondrá la posibilidad de realizar las sesiones de entrenamiento de una forma mucho más eficiente, lo que posibilita el tratamiento de un número de clientes mayor y como resultado, un mayor beneficio económico. Además, de los clientes que no tienen posibilidad de acudir a un centro de rehabilitación de forma frecuente porque se encuentren en zonas rurales o alejadas de los centros médicos.

### 6.1 Trabajo futuro

Para el trabajo futuro, como se ha mencionado anteriormente, se propone la posibilidad de añadir nuevos modos en los que se realiza el movimiento del avatar. La aplicación ha sido desarrollada de tal modo que la adición de esta parte del software no afectará al funcionamiento del resto de la aplicación.

Actualmente existen numerosas alternativas para realizar el tracking o seguimiento del cuerpo para aplicaciones de realidad virtual, una de ellas es *Antilatency tracker* [2], un pequeño dispositivo que se puede integrar en los visores de realidad virtual actuales para conseguir un seguimiento completo del movimiento de todo el cuerpo del usuario.



**Figura 39. Antilatency tracker**

Mediante el uso de este dispositivo se puede conseguir realizar el movimiento del avatar de una manera mucho más exacta y cómoda para el fisioterapeuta. Como trabajo futuro se propone el uso de este dispositivo, adaptando la parte del movimiento del avatar a este nuevo sistema. El principal problema del uso de este sistema es el coste, ya que solamente el dispositivo tiene un coste de unos 650\$, sin tener en cuenta licencias y otros dispositivos que son necesarios para hacerlo funcionar.

Otra alternativa para mejorar el funcionamiento de la aplicación podría consistir en realizar el tracking del cuerpo del fisioterapeuta mediante el uso de una cámara. Existen numerosos plugins implementados actualmente para Unity que permiten realizar el seguimiento mediante estos sistemas, uno de ellos es *OpenPose* [10], que permite registrar el movimiento del cuerpo humano mediante la detección de las diferentes articulaciones que componen el cuerpo.

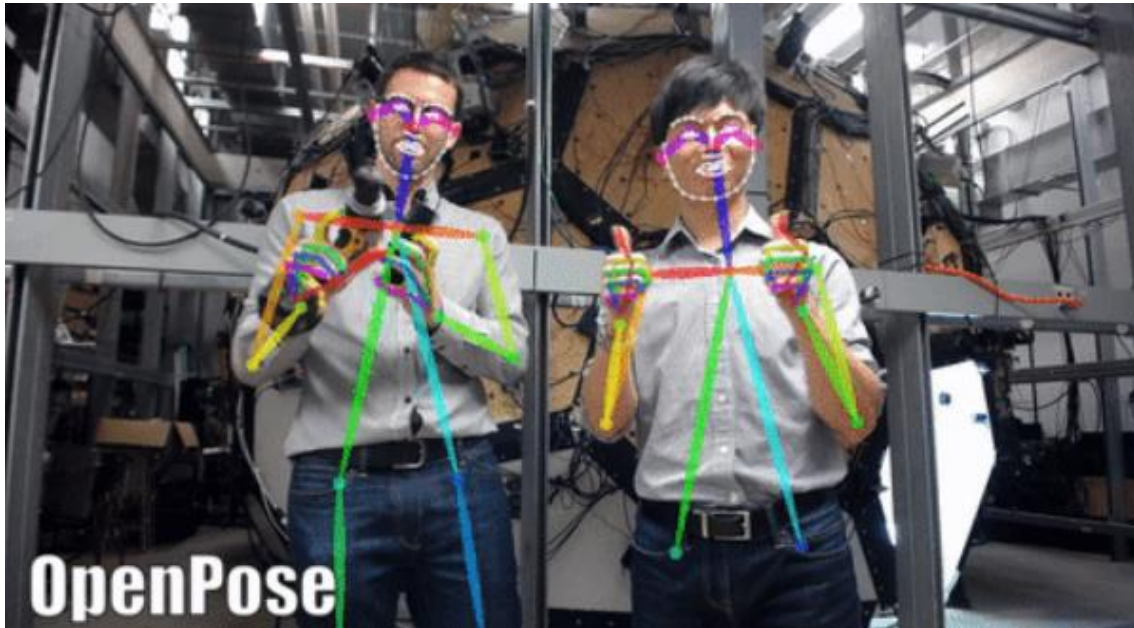


Figura 40. OpenPose

Aparte del desarrollo de nuevos sistemas para el movimiento del avatar, se propone como trabajo futuro el desarrollo de una aplicación superior que permita el acople de los 2 módulos desarrollados en una única aplicación para comodidad del usuario final. Esta aplicación final contemplaría además el uso de una interfaz gráfica y diferentes manuales para facilitar el proceso al fisioterapeuta. El uso de esta interfaz gráfica, también posibilita la modificación de algunos parámetros que no se han barajado para este proyecto, como modificar el tiempo de descanso existente entre los diferentes ejercicios de la sesión de entrenamiento.

La aplicación superior también podría contener la funcionalidad de envío de las sesiones de entrenamiento creadas desde la aplicación del terapeuta a la aplicación del paciente. Esta funcionalidad contemplaría el envío del fichero JSON, el cual tiene un tamaño muy reducido, a través de una conexión a internet. De esta manera, se automatiza el proceso de comunicación entre las aplicaciones terapeuta-cliente.

Se propone además la mejora de la aplicación del paciente, de forma que sea capaz de recoger diferentes datos en la ejecución de las sesiones de entrenamiento por parte del paciente y de esta forma mediante el análisis de estos datos, poder mejorar las sesiones de entrenamiento de forma más personalizada para cada paciente. Estos datos podrían incluir por ejemplo el tiempo en realizar las sesiones o los ejercicios en los que el paciente ha tenido mayor dificultad.

Como punto final para trabajos futuros, aprovechando el gran potencial que ofrece Unity, se podría lanzar la aplicación del proyecto en diferentes sistemas, como en dispositivos Android, dispositivos de realidad virtual e incluso en consolas. Para ello, simplemente sería necesario adaptar la entrada a las diferentes plataformas y mediante Unity desplegar la aplicación a la plataforma deseada.



## Capítulo 7. Bibliografía

- [1] Y. M. Anoffo, «DEVELOPMENT OF A VIRTUAL ENVIRONMENT FOR FULL-BODY MOTOR TRAINING. Trabajo final de máster MUIT - ETSIT UPV,» Valencia, 2019.
- [2] «Dispositivos hápticos en la Realidad Virtual,» [En línea]. Available: <https://quasardynamics.com/dispositivos-hapticos-la-realidad-virtual/>.
- [3] «Realidad Virtual: otro mundo al alcance de tus ojos,» [En línea]. Available: <https://www.iberdrola.com/innovacion/realidad-virtual>.
- [4] «Rehabilitación virtual,» [En línea]. Available: <https://rehametrics.com/rehabilitacion-virtual-juegos/>.
- [5] M. Pedraza-Huesoa, S. Martín-Calzón, F. J. Díaz-Pernas y M. Martínez-Zarzuela, «Rehabilitation using Kinect-based Games and Virtual Reality,» 2015.
- [6] D. Antón, A. Goñi, A. Illarramendi, J. J. Torres-Unda y J. Seco, «KiReS: A Kinect-based telerehabilitation system».
- [7] E. Shabnam Sadeghi, T. Thompson, A. Parsa, I. Browna y S. Cirstea, «ReHabgame: A non-immersive virtual reality rehabilitation system with applications in neuro science,» Cambridge, 2017.
- [8] «Cinemática inversa,» [En línea]. Available: [https://www.ecured.cu/Cinem%C3%A1tica\\_Inversa](https://www.ecured.cu/Cinem%C3%A1tica_Inversa).
- [9] «Inverse Kinematics (Cinemática Inversa),» [En línea]. Available: <https://docs.unity3d.com/es/2019.4/Manual/InverseKinematics.html>.
- [10] «OpenPose,» [En línea]. Available: <https://github.com/CMU-Perceptual-Computing-Lab/openpose>.
- [11] «Realidad virtual,» [En línea]. Available: <http://mundo-virtual.com/que-es-la-realidad-virtual/>.
- [12] «Unity,» [En línea]. Available: <https://unity.com/es>.
- [13] «MakeHuman,» [En línea]. Available: [http://www.makehumancommunity.org/content/makehuman\\_120.html](http://www.makehumancommunity.org/content/makehuman_120.html).
- [14] «YVTuber,» [En línea]. Available: <https://github.com/youten/YVTuber>.
- [15] «Visual studio,» [En línea]. Available: <https://visualstudio.microsoft.com/es/>.
- [16] «Antilatency tracker,» [En línea]. Available: <https://antilatency.com/>.
- [17] «Introducción al desarrollo de videojuegos en Unity,» [En línea]. Available: <https://www.edx.org/course/introduccion-al-desarrollo-de-videojuegos-con-unit>.
- [18] «Introducción al desarrollo de videojuegos con Unity 3D,» [En línea]. Available: <http://repositori.uji.es/xmlui/handle/10234/170727>.
- [19] «Colliders detección de colisiones en Unity,» [En línea]. Available: <https://academiaandroid.com/colliders-deteccion-colisiones-juego-unity-3d/>.



- [20] M. K. HOLDEN, «Virtual Environments for Motor Rehabilitation: Review,» 2005.
- [21] M. García y F. Ricardo, «Cinemática inversa,» [En línea]. Available: [http://www.eudim.uta.cl/rmendozag/courses/fundamentos\\_robotica/lectures/cinematica\\_inversa.pdf](http://www.eudim.uta.cl/rmendozag/courses/fundamentos_robotica/lectures/cinematica_inversa.pdf).
- [22] P. Lista, D. González y C. Souto, «¿Qué papel desempeña la Fisioterapia en la pandemia mundial por COVID-19?».