

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

Grado en Ing. Sist. de Telecom., Sonido e Imagen

---



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



ESCUELA POLITECNICA  
SUPERIOR DE GANDIA

**“Diseño y desarrollo de una  
aplicación de control de ejercicios  
de fisioterapia mediante el uso del  
dispositivo Kinect V2”**

**TRABAJO FINAL DE GRADO**

Autor/a:

**Rafael Conejero Vila**

Tutor:

**Fernando Boronat Seguí**

**GANDIA, 2017**

## RESUMEN

Se presenta una aplicación diseñada para el control de ejercicios de fisioterapia mediante el uso del dispositivo Kinect V2. Esta aplicación será utilizada por un especialista de la fisioterapia y por los pacientes. Por parte del fisioterapeuta, permite diseñar ejercicios de rehabilitación, y gestionar y controlar mediante una base de datos: los pacientes involucrados, las sesiones a realizar y los ejercicios programados para cada sesión. Por parte del paciente, agiliza la realización de las sesiones programadas que contienen ejercicios en un entorno 3D controlados por el dispositivo Kinect V2, y proporciona acceso a su historial sobre las sesiones realizadas. La versión diseñada consta de tres ejercicios para la rehabilitación de la articulación glenohumeral (hombro). El objetivo principal es facilitar el seguimiento individualizado de la rehabilitación que cada fisioterapeuta tiene con su paciente. De esta forma, el paciente podrá volver a su vida anterior a la lesión en el menor tiempo posible, ya que este seguimiento permite que ambos puedan avanzar conjuntamente hacia la recuperación de la salud del paciente.

## ABSTRACT

In this project, an application for the control of physiotherapy exercises through the use of the Kinect V2 device is presented. The application will be used by a physiotherapy specialist and by the patients. On the part for the physiotherapist, it allows to design rehabilitation exercises, and manage and control them through a database: the patients involved, the sessions to be performed and the exercises scheduled for each session. On the part for the patient, it facilitates the performance of the scheduled sessions that contain exercises in a 3D environment controlled by the device Kinect V2, and provides access to their history about the realized sessions. The designed version consists of three exercises for the rehabilitation of the glenohumeral joint (shoulder). The main objective is to facilitate the individualized follow-up of the rehabilitation that each physiotherapist has with his/her patient. This way, the patient will be able to return to his/her life prior to the injury in the shortest possible time, since this monitoring allows both to progress together towards the recovery of the patient's health.

**Palabras clave:** Fisioterapia, Kinect V2, Control de gestos, Base de datos, C#.

**Keywords:** Physiotherapy, Kinect V2, Gesture control, Database, C#.

---

# INDICE

<b>CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS DEL PROYECTO.....</b>	<b>4</b>
1.1. INTRODUCCIÓN.....	4
1.2. ANTECEDENTES.....	4
1.3. OBJETIVOS DEL PROYECTO .....	5
1.4. ESTRUCTURA DE LA MEMORIA.....	6
<b>CAPÍTULO 2: HISTORIA DEL ARTE.....</b>	<b>8</b>
2.1. BIO-CIRAC .....	8
2.2. APLICACIÓN PARA NIÑOS AUTISTAS.....	9
2.3. APLICACIÓN PARA REHABILITACIÓN DE MIEMBROS SUPERIORES.....	9
2.4. VITA.....	10
2.5. APP'S PARA MÓVILES .....	11
<b>CAPÍTULO 3. HERRAMIENTAS UTILIZADAS .....</b>	<b>13</b>
3.1. HERRAMIENTAS HARDWARE .....	13
3.2. HERRAMIENTAS SOFTWARE.....	15
<b>CAPÍTULO 4. DESARROLLO DE LA APLICACIÓN .....</b>	<b>19</b>
4.1. INSTALACIÓN Y CONFIGURACIÓN DEL DISPOSITIVO KINECT V2 .....	19
4.2. INSTALACIÓN DE LA APLICACIÓN NI-MATE 2.....	20
4.3. INSTALACIÓN Y CONFIGURACIÓN DEL PLUG-IN FOR NI-MATE DE BLENDER .....	20
4.4. DESARROLLO DE LA APLICACIÓN TRIDIMENSIONAL EN BLENDER .....	21
4.5. BASE DE DATOS MYSQL .....	38
4.6. APLICACIÓN GRÁFICA DE USUARIO Y DE ACCESO A LA BASE DE DATOS	40
<b>CAPÍTULO 5. CONCLUSIONES .....</b>	<b>49</b>
<b>CAPITULO 6. BIBLIOGRAFÍA.....</b>	<b>50</b>

---

# CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS DEL PROYECTO

## 1.1. INTRODUCCIÓN

Tanto los pacientes con una lesión musculoesquelética como los pacientes con enfermedad física crónica, necesitan seguir un tratamiento para su recuperación. Los primeros lo necesitan para recuperar su movilidad fisiológica mientras su lesión vaya sanando, y los segundos para mejorar su calidad de vida.

Para ello, los fisioterapeutas, como profesionales sanitarios, ofrecen alternativas terapéuticas no farmacológicas; siendo los más cualificados para este tipo de tratamientos [1-FISIO]. Deben tratar todo tipo de lesiones y dolores musculoesqueléticos que cualquier paciente pueda tener, y, de esta forma, ayudar a que puedan volver a su estado fisiológico anterior a la lesión o dolor. Un fisioterapeuta está capacitado para aplicar este tipo de tratamientos, realizar ejercicios de rehabilitación con el paciente y ayudarlo en todo lo necesario para una recuperación individualizada y adecuada.

Muchos tratamientos, cuando llegan al final de las sesiones programadas, no requieren de un fisioterapeuta supervisor, ya que es el propio paciente el que puede y debe realizar en la consulta, o incluso en su domicilio, los ejercicios programados. Sin embargo, en algunos casos, puede que el paciente no realice los ejercicios o que los realice de forma incorrecta. En este proyecto se presenta una aplicación de rehabilitación interactiva que mejora la experiencia del paciente en su tratamiento (en su domicilio o en la consulta) y que resulta beneficiosa psicofisiológicamente. Además, la aplicación desarrollada permite al fisioterapeuta realizar una gestión y control más prácticos y completos las sesiones de sus pacientes.

El presente proyecto, en términos generales, consiste en realizar una aplicación que sea de utilidad tanto para el paciente como para el fisioterapeuta. Por el lado del paciente consistirá en una aplicación para realizar los ejercicios sin requerir la presencia del fisioterapeuta. Ello se conseguirá mediante un control utilizando un dispositivo de reconocimiento de movimientos corporales, como es el dispositivo Kinect V2 de Microsoft (a partir de ahora Kinect V2). Por el lado del sanitario, consistirá en una aplicación que le permita gestionar, programar y revisar sesiones que sus pacientes deberán ir realizando, así como también revisar a posteriori las que ya hayan finalizado.

## 1.2. ANTECEDENTES

El grupo de investigación Immersive Interactive Media (IIM) R&D Group, de la Universitat Politècnica de València (a partir de ahora UPV) en el Campus de Gandía, dispone de varios dispositivos para capturar el movimiento, entre los que se destacan: una sala de captura de movimientos (MoCap o Motion Capture Lab), donde mediante 16 sensores y un traje especial, detecta perfectamente la posición y movimiento de una persona; dos dispositivos Kinect V2, que a través de sus elementos de detección reconocen los usuarios que se coloquen delante así como sus movimientos; y dos dispositivos Leap Motion para control de los movimientos de las manos y antebrazos.

Se propuso realizar una aplicación utilizando uno de estos dispositivos, pero al pensar en una posible implementación comercial, se decidió utilizar el dispositivo Kinect V2, por ser más práctico y económico.

Para realizar la aplicación que será utilizada por un fisioterapeuta y sus pacientes, uno de los puntos fuertes es reunirse con un fisioterapeuta, para que explique qué necesidades tiene que cubrir el programa que se pretende desarrollar. Para ello, mediante la ayuda de un miembro del grupo de investigación de la UPV, se contactó con un fisioterapeuta colegiado con una consulta en la ciudad de Gandía. Se acordó los datos a almacenar en una base de datos sobre los pacientes, sus sesiones y los ejercicios para rehabilitación del hombro.

### 1.3. OBJETIVOS DEL PROYECTO

El objetivo principal del proyecto consiste en diseñar y desarrollar una aplicación basada en control de gestos mediante el dispositivo KinectV2, que contenga una parte de gestión administrativa y una parte de ejecución interactiva con un entorno gráfico tridimensional de ejercicios fisioterapéuticos. Dicha aplicación debe ser autónoma, sin necesidad de la presencia física de un fisioterapeuta controlando la sesión de rehabilitación.

El objetivo principal se puede dividir en diversos sub-objetivos:

- a) Desarrollar un módulo de conexión entre el dispositivo Kinect V2 y un entorno gráfico tridimensional (en 3D).
- b) Desarrollar un módulo de control de un avatar que represente los movimientos del paciente.
- c) Crear un entorno 3D que incluya varios ejercicios de rehabilitación.
- d) Crear una base de datos con los datos de los pacientes, fisioterapeutas, ejercicios y sesiones. Deberá ser relacional, con las tablas y conexiones necesarias entre ellas.
- e) Crear una interface gráfica (GUI) para acceder a dicha base de datos con todos los formularios pertinentes.
- f) Crear una aplicación integrada que englobe la GUI con el entorno 3D de realización de ejercicios controlados por el dispositivo Kinect V2 e interactúe con la bases de datos.
- g) Generar un archivo ejecutable para instalar la aplicación integrada en las dependencias del usuario (fisioterapeuta o paciente).

Los ejercicios de rehabilitación a incluir en esta primera versión de la aplicación están pensados para rehabilitación de la articulación glenohumeral (más conocida como hombro). Los ejercicios escogidos por el fisioterapeuta fueron los tres siguientes:

- 1) Abducción: Levantar el brazo lateralmente

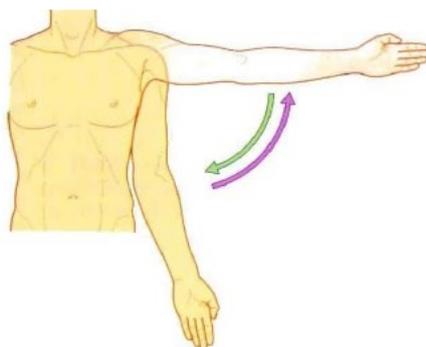
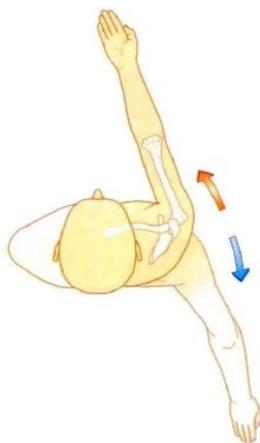


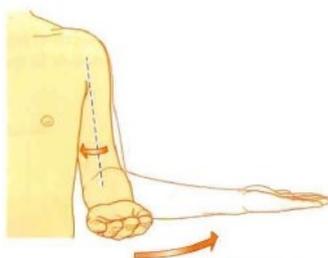
IMAGEN 1: ABDUCCIÓN GLENOHUMERAL.

2) Flexión: Movimiento del brazo hacia delante y hacia atrás



**IMAGEN 2: FLEXIÓN GLENOHUMERAL.**

3) Rotación: Habiendo 90° entre el brazo y el antebrazo, movimiento del brazo apuntando hacia la derecha y hacia la izquierda



**IMAGEN 3: ROTACIÓN GLENOHUMERAL.**

Un ejemplo gráfico de la disposición de los diferentes elementos. El paciente se situará delante de la pantalla, entre 1,5 y 3 metros, donde se encuentra el dispositivo Kinect V2.



**IMAGEN 4: DISPOSICIÓN DE LOS DISTINTOS ELEMENTOS.**

## 1.4. ESTRUCTURA DE LA MEMORIA

En el Capítulo 2 se presentan algunos proyectos y aplicaciones relacionadas con este proyecto. Se presentan cuatro proyectos con contenido interactivo, desde aplicaciones con el dispositivo Kinect V2 como dispositivo de captación hasta dispositivos

específicos. Además, se presentan aplicaciones para dispositivos móviles para ayudar a realizar ejercicios y rehabilitación de lesiones.

En el capítulo 3 se dispone de una explicación detallada de las herramientas utilizadas, tanto hardware como software.

El capítulo 4 contiene la explicación del proceso de crear la aplicación final, desde la obtención de los datos a través del dispositivo Kinect V2 hasta el entorno visual que será utilizado por los usuarios.

Además, se incluyen los siguientes anexos:

1. Configurar el programa Ni-Mate para el funcionamiento de la aplicación.
2. Un manual de usuario para saber cómo funciona la aplicación.
3. Como crear un formulario en visual C# desde cero.
4. Como crear una aplicación en Blender con el dispositivo Kinect V2.
5. Como crear una animación en Blender.
6. Instalar el plug-in de Ni-Mate en Blender.

# CAPÍTULO 2: HISTORIA DEL ARTE

En este capítulo se presentan tesis, trabajos universitarios y aplicaciones para dispositivos móviles encontrados a través de la biblioteca interactiva de la Universidad Politécnica de Valencia y la plataforma de búsqueda Scholar de Google que, aun siendo para el mismo o distinto propósito, se ve en interés por instaurar la tecnología en el ámbito médico no invasivo.

## 2.1. BIO-CIRAC

Esta aplicación fue desarrollada por J.F Villada, J. E. Muñoz y J.D Hoyos en Colombia, segundo semestre de 2014, y consiste en un software de análisis biomecánico a través del dispositivo Kinect como herramienta la cual permite controlar e interactuar ambientes virtuales sin necesidad de utilizar una interfaz táctil y permitiendo una interacción sin contacto con el paciente. Con esto, capturar en tiempo real las posiciones y ángulos de las articulaciones. Realiza el análisis biomecánico del movimiento, monitoriza en tiempo real el proceso de rehabilitación y cuantifica de los resultados<sup>1</sup>.

Utiliza librerías de Matlab que fueron desarrollados por Neil Lawrence que permite la extracción de los datos capturados.

En la siguiente imagen se muestra el interfaz GUI del software desarrollado.

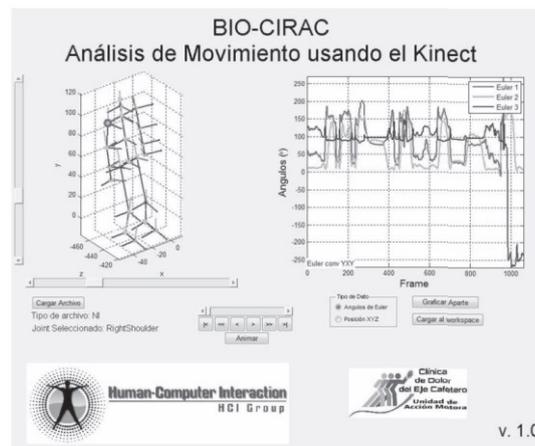


IMAGEN 5: INTERFAZ GUI DEL SOFTWARE BIO-CIRAC.

Con esta aplicación se mejora la motivación del paciente por tener la oportunidad de poder monitorear dinámicamente variables fisiológicas mientras realiza los ejercicios prácticos para su recuperación, además de poder complementar con datos biomecánicos el historial del paciente.

El aporte de esta aplicación descrita es de conocer alternativas para la captura de movimientos a través del dispositivo Kinect como el uso de librerías en la herramienta Matlab. La aplicación desarrollada en este TFG, mejora la interacción con el paciente con los ejercicios a realizar, además, el fisioterapeuta puede tener centralizado todos los pacientes, en una base de datos.

<sup>1</sup> Entre Ciencia e Ingeniería. Dec. 2014 Vol. 8 Issue 16, p72-77 6p. Autores: Villada, J. F.; Muñoz, J. E.; Hoyos, J. D. Web.

<http://biblioteca.ucp.edu.co/ojs/index.php/entrecei/article/viewFile/2499/2380>

## 2.2. APLICACIÓN PARA NIÑOS AUTISTAS

Plataforma interactiva que hace uso del dispositivo Kinect aplicada al tratamiento de niños autistas. Esta aplicación, aun no siendo para la rehabilitación física, si realiza un tratamiento mediante el dispositivo NUI (interfaz natural de usuarios) que se explicará más adelante) para la mejora de niños con problemas de autismo<sup>2</sup>. Esta plataforma contiene siete aplicaciones para niños que padezcan trastorno motriz espacial y sensorial, otros trastornos como auditivo, trastorno de conceptualización. Mediante un entorno visual y la reproducción de audio, los pacientes reciben las instrucciones a realizar en cada aplicación.

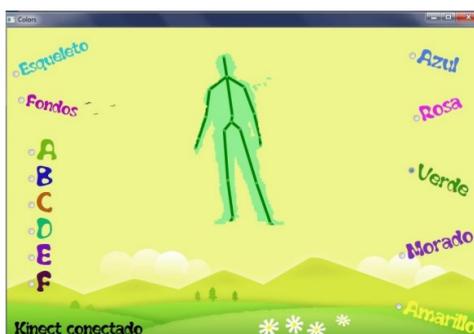


IMAGEN 6: APLICACIÓN PARA UNIR LETRAS DE COLORES CON LOS NOMBRES DEL COLOR.

Cumple con los objetivos al ser una herramienta de apoyo en las terapias de los niños con autismo. Además, puede ser tomado para niños con otros trastornos específicos que trate esta aplicación. Y con esto, mejorar la calidad de las terapias.

Es otra aplicación ejemplo con el uso del dispositivo Kinect, en esta aplicación aporta la interacción del paciente y el entorno creado especialmente para su fin. La mejora de la aplicación de este TFG es la de disponer de una base de datos con la información de todos los pacientes y su evolución.

## 2.3. APLICACIÓN PARA REHABILITACIÓN DE MIEMBROS SUPERIORES

Proyecto realizado en la universidad de Caracas, Venezuela, para la rehabilitación mediante movimientos físicos del paciente, que consiste en un framework para la rehabilitación física en miembros superiores con realidad virtual. Dispone de distintos juegos en los que el paciente debe de entrar para completar su tratamiento<sup>3</sup>.



IMAGEN 7: APLICACIÓN PARA RECOGER VIRTUALMENTE OBJETOS Y COLOCARLOS DONDE CORRESPONDE.

<sup>2</sup> Escuela superior de Ingeniería mecánica y eléctrica. México. Autores: Cruz Lara, R.; Fernández Fabela, H. F.; Olvera Morales, A. Web.

<http://tesis.ipn.mx/bitstream/handle/123456789/12504/PLATAFORMAKINECT.pdf>

<sup>3</sup> CoNCISa (2013) Sección 3. Autores: F. Moreno, J. Ojeda. E. Ramírez, C. Mena, O. Rodríguez Web. <http://lcg.ciens.ucv.ve/~esmitt/publications/2013/concisa13.pdf>

Este framework consiste en un sistema de realidad virtual de bajo coste para uso clínico, para pacientes entre 6 y 12 años con discapacidades en miembros superiores. Se trata de un sistema que no representa ningún riesgo para el paciente.

Este proyecto centrado en la rehabilitación, aporta a la aplicación de este TFG los ejercicios específicos para los miembros superiores del cuerpo. Las mejoras que se aplican es la base de datos de los pacientes, además de resultar de una aplicación más seria centrada más en la rehabilitación de la zona afectada con los ejercicios sugeridos por un fisioterapeuta.

## 2.4. VITA

Interfaz de apoyo para la rehabilitación física de personas con osteoartritis de rodilla por obesidad, en un sistema integral para una continuidad del tratamiento. Esta interfaz está diseñada para realizar un tratamiento que conlleva la combinación de terapias físicas y/u ocupacionales, variedad de ejercicios, control del peso, educación del paciente y medicamentos<sup>4</sup>.

Contiene 4 elementos, la interfaz electrónica, dos elementos para la rodilla afectada que son una rodillera y unos elementos térmicos para aplicar frío/calor, un manual de ejercicios y la propia aplicación. Se puede considerar como una interfaz interactiva adaptable, llevando un equilibrio entre el ejercicio y el descanso de un tratamiento de rehabilitación física. Esta interfaz promoverá a los pacientes la continuidad en la realización de la terapia pudiendo ser llevado a un entorno médico o en el hogar.



IMAGEN 8: ELEMENTOS QUE CONTIENE LA INTERFAZ.

Con esta interfaz, tanto el especialista como el paciente tienen acceso a la evolución del tratamiento. Además, éste último, recibe mensajes recordatorios del tratamiento que debe seguir y lo anima a intentar superarlo.

Esta aplicación no utiliza el dispositivo Kinect, no obstante el uso de un sistema propio para la rehabilitación de la articulación de la rodilla, aporta la necesidad de aplicaciones para la mejora de los pacientes.

<sup>4</sup> Universidad ICESI (2013) Autores: A. Montoya Montoya, D. C. Orejuela V. Web. <https://repository.icesi.edu.co>

## 2.5. APP'S PARA MÓVILES

También cabe destacar las aplicaciones para terminales móviles, que ya sea como recordatorios, guías, o incluso utilizando los sensores propios de los Smartphone, pueden ayudar a la rehabilitación de los pacientes.

### 2.5.1. GONIOMETER

Aplicación para móvil o Tablet que disponga de un inclinómetro de gravedad integrado. Fue diseñada con el propósito de desarrollar una herramienta confiable y fácil de usar para la medición del rango de movimiento. Se ha sometido a pruebas con éxito en ortopedia, quiropráctica y fisioterapia. Puede también utilizarse en los ámbitos de la medicina deportiva, la rehabilitación, la radiología, la ingeniería, la arquitectura y otras ramas.

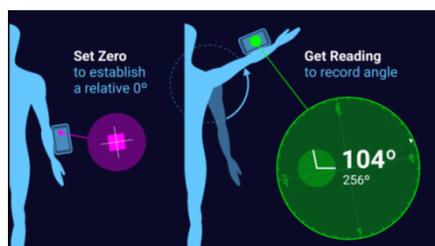


IMAGEN 9: COMO UTILIZAR LA APLICACIÓN GONIOMETER.

Su funcionamiento es el siguiente: colocar el dispositivo sujeto a la articulación; establecer el 0° como punto cero relativo; realizar el movimiento del dispositivo hasta el ángulo deseado y obtener la lectura. Puede memorizar hasta doce mediciones distintas.

Esta aplicación es una alternativa a la detección de los movimientos del cuerpo a través de Kinect. Mejora con respecto a la movilidad del sistema, pero no puede realizar un control sin errores como el caso del reconocimiento a través del dispositivo Kinect V2.

### 2.5.2. PHYSIOADVISOR EXERCISES

Proporciona al usuario una amplia gama de ejercicios para mejorar la flexibilidad de la fuerza, la condición física y la postura. El contenido de esta App fue creado por fisioterapeutas experimentados. Proporciona instrucciones con imágenes claras para un fácil entendimiento para el usuario. Incluye una tabla de diversas zonas del cuerpo, que permite a los usuarios hacer referencia a determinadas partes y encontrar ejercicios específicos para esa área. Es una opción para crear un programa de rehabilitación con una función recordatorio. Además, se pueden crear programas para la prevención de lesiones.



IMAGEN 10: SELECCIÓN DE UNA PARTE DEL CUERPO PARA MOSTRAR LOS EJERCICIOS ESPECIFICOS.

Esta aplicación aporta al TFG los aspectos relevantes del software en la aplicación móvil que acompaña a los usuarios y les da retroalimentación del proceso de mejoramiento de su postura, además de llevar un registro día a día que les permite comparar su desarrollo.

### 2.5.3. REHABILITATION FOR LOWER LIMBS

Permite a los fisioterapeutas y profesionales de la salud comunicar eficazmente la información sobre lesiones a sus pacientes y que puedan entender mejor el ejercicio a realizar. Contiene una lista completa de 165 lesiones, condiciones y trastornos que pueden afectar a los miembros inferiores del cuerpo.

Permite al usuario seleccionar lesiones, y ver un modelo detallado de la estructura anatómica, mostrando una descripción de la lesión y una lista de todas las otras lesiones relacionadas con esta área del cuerpo.

El usuario puede elegir entre todos los ejercicios de la lista clínicamente aplicables para esa lesión en particular.



IMAGEN 11: INFORMACIÓN PROPORCIONADA POR LA APLICACIÓN.

Esta aplicación aporta al TFG una visión de alternativas para presentar herramientas para la ayuda a la rehabilitación de los pacientes.

---

# CAPÍTULO 3. HERRAMIENTAS UTILIZADAS

En la aplicación desarrollada se necesita una parte física o hardware, y una parte virtual o software imprescindibles para el correcto funcionamiento de la misma.

Serán necesarios dos dispositivos hardware inseparables: un ordenador, con todos sus componentes necesarios (más adelante se especificarán), y un dispositivo de interacción natural (interfaz natural de usuarios o NUI), como es el dispositivo KinectV2 de Microsoft.

Durante la historia, ha habido 3 tipos de interacciones de usuarios: CLI, GUI, y NUI. [2-WIKI]

- CLI (interfaz de línea de comandos) es un método donde el usuario puede dar instrucciones a un programa informático mediante líneas de texto simple. Debe de utilizar unos códigos específicos para que dicho programa entienda y funcione de la forma que quiere el usuario. Principalmente se utiliza como parte física un teclado y una pantalla. Con el teclado el usuario interactúa con el programa, y con la pantalla el programa interactúa con el usuario.
- GUI (interfaz gráfica de usuarios) se trata de un software que, utilizando los mismos dispositivos físicos que en el caso del CLI y añadiendo otros como un ratón, un Gamepad y otros muchos; mediante imágenes y elementos intuitivos, un usuario puede interactuar de forma más cómoda con el programa. Para ello, ya se requiere de un sistema mucho más robusto y dinámico que un simple programa. Los Sistemas Operativos son uno de ellos, entre otros posibles.
- NUI (interfaz natural de usuarios), es un interfaz como el GUI. La diferencia entre ambos es que, donde antes el usuario requería de un ratón o un teclado para interactuar con el software, ahora solo requiere de movimientos gestuales del cuerpo, tales como movimientos de las manos, gesticulaciones de partes de la cara, control mediante la voz, etc.

## 3.1. HERRAMIENTAS HARDWARE

El sistema NUI que se va a utilizar, requiere de un dispositivo externo que irá conectado al ordenador. Este dispositivo será el encargado de detectar al usuario para poder controlar mediante sus movimientos el programa. Se trata del dispositivo Kinect V2.

El dispositivo Kinect V2 es un dispositivo controlador creado por Alex Kipman en el año 2010 y desarrollado por Microsoft para sus consolas Xbox. Inicialmente fue pensado para su uso exclusivo como controlador de juegos, pero gracias a su gran capacidad, muchos desarrolladores empezaron a implementar software de interacción con elementos virtuales a través de los movimientos del cuerpo humano. Por ello, Microsoft, en un primer momento, permitió que desarrolladores de aplicaciones pudiesen conseguir el SDK (kit de desarrollo de software) para crear otras aplicaciones, además de los juegos para consolas, y, más adelante, también permitió que cualquier usuario pudiese acceder a dicho SDK. [3-WIKI]

Para esta aplicación se ha optado por utilizar el dispositivo Kinect versión "V2.0 for Windows". Trae distintas actualizaciones con respecto al primer dispositivo que sacó la

compañía de Microsoft. Como se muestra en la imagen 12, las características principales son las siguientes: cuenta con una cámara RGB, una matriz de emisores infrarrojos (IR) y cámara IR, así como también una matriz de micrófonos.

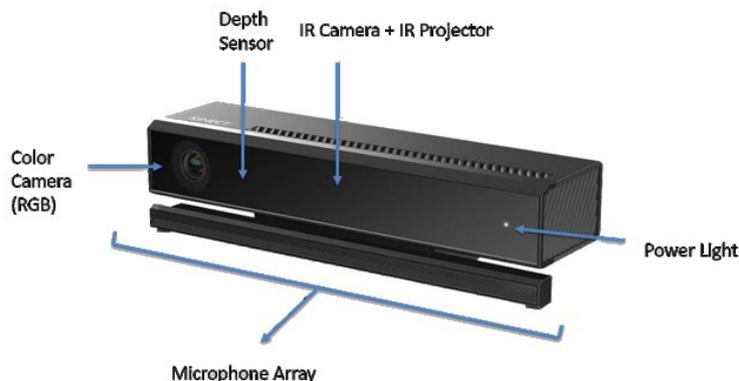
La cámara RGB (Red, Green, Blue) se utiliza para la captación del entorno, los colores y su forma.

La matriz IR es la encargada de localizar los objetos en el espacio. Para ello, utiliza dos técnicas:

- 1) Cuanto más cerca está un objeto del emisor IR, el brillo de la reflexión determina la proximidad de dicho objeto.
- 2) La ubicación del emisor IR y de la cámara IR es distinta por tanto a un lado del objeto se crea una sombra que dependiendo de la proximidad que haya entre objetos será más o menos grande.

De estas formas, se consigue obtener con muy buena precisión la posición y forma de cada objeto que se encuentre delante. Además, con la imagen captada por la cámara RGB, se puede crear con facilidad una simulación 3D del objeto con sus colores y demás elementos. Otra ventaja, es que al consistir de varios emisores y la cámara IR, no necesita iluminación externa para un correcto funcionamiento.

Por último, la matriz de micrófonos puede conseguir localizar el foco que emite el sonido y cancelar el ruido externo, así como la dirección de su emisión.



**IMAGEN 12: DISPOSICIÓN DE ELEMENTOS DEL DISPOSITIVO KINECT V2.**

Todo lo anterior no sería posible sin la capacidad del procesador interno que incorpora el dispositivo Kinect V2. Éste es el encargado de detectar a las personas (hasta 6 usuarios simultáneamente), los gestos de la cara, detectar la voz principal y anular todos los sonidos restantes, y otras funciones más. Hacia el ordenador, sólo se envía la información ya procesada. De esta forma, no se sobrecarga el procesador del ordenador realizando las tareas de detección.

Por otro lado, el ordenador necesario para poder utilizar correctamente el dispositivo Kinect V2 descrita, requiere de unas características específicas mínimas: Un procesador de doble núcleo físico de 3,1 GHz a 64 bits, disponer de un controlador y un bus dedicado USB 3.0, una memoria RAM de 4 GB, y una tarjeta gráfica que admita DirectX 11 gama media.

En el caso del presente proyecto, se ha utilizado un ordenador con un procesador i7 de Intel de 4 núcleos a 3,4 GHz y de 64 bits, la propia placa base dispone de puertos USB 3.0, 16 GB de RAM DDR3 y una tarjeta gráfica, muy superior a la mínima recomendada, marca GeForce modelo GTX 970. Con esta composición, el equipo no

ha tenido ningún problema en procesar en tiempo real los programas que se le han instalado así como los desarrollados en el proyecto.

## **3.2. HERRAMIENTAS SOFTWARE**

En este apartado se va a diferenciar entre el software necesario para el funcionamiento de la aplicación y el necesario para la creación de la misma.

### **3.2.1. SISTEMA OPERATIVO**

Para empezar, todo ordenador personal necesita de un Sistema Operativo (a partir de ahora SO), el cual es el encargado de gestionar los recursos de todos los elementos de hardware y software instalados. Para la elección de cuál es el mejor SO para la aplicación del proyecto, se tuvo en cuenta los requisitos mínimos que exigía la utilización del dispositivo Kinect V2, que son un SO Windows 8 o superior [4-KINECT]. Para este caso, se utilizó la versión de Enterprise de 64Bits de Windows 10. Tanto la aplicación de este proyecto como dispositivo Kinect V2 utilizado, pueden ser utilizados con la versión de Windows 8.

Una vez decidido el SO a utilizar, es cuando se deben instalar todas las aplicaciones para que funcione todo el sistema correctamente. En primer lugar, los drivers (software dedicado al control y funcionamiento de dispositivos externos) del dispositivo Kinect V2 y su SDK, el servidor que gestionará la base de datos MySQL, los conectores necesarios para el acceso a la base de datos (para lenguaje C# y Python), y un programa intermedio llamado Ni Mate de la empresa Delicocode, que se encarga de recibir los datos que obtiene. Sólo con estos programas será suficiente para el buen funcionamiento de la aplicación del proyecto. Para la creación de los programas finales, se ha utilizado diversos software. Se ha utilizado la herramienta Blender, para la creación del programa 3D, MySQL Workbench para la creación y gestión de la base de datos y, por último, Visual Studio para la creación del entorno visual de la base de datos.

### **3.2.2. BASE DE DATOS**

Para la creación de la base de datos, en primer lugar se tuvo en consideración todas las opciones ya existentes. Unas posibles opciones son: programas que ya realizan dicha función como "Access" de Microsoft; versiones gratuitas como "Base" de Openoffice. "Access" de Microsoft dispone de una extensión que transfiere la base de datos y el motor de gestión a un archivo ejecutable; mientras que "Base" de Openoffice, la única solución encontrada es utilizar una versión autoejecutable de Openoffice para la NO instalación del programa "simulando" un programa ejecutable.

Por otro lado, respecto a las licencias, "Access" es de pago, y no se dispone de licencia. En el caso de "Base", es totalmente gratuito y libre, por tanto, no habría problemas con la autolimitación impuesta para la creación de la aplicación final. Con los inconvenientes de uno y otro, se acuerda investigar con respecto a las bases de datos SQL<sup>5</sup>.

Al tener pocos conocimientos de "SQL", se encontró una herramienta gratuita y de programación gráfica llamada "MySQL". Con ésta, se pueden crear y gestionar bases de datos a gran escala. La característica más atractiva es la posibilidad de acceder a la base de datos a través de internet y poderse utilizar para páginas webs u otras aplicaciones con acceso a la red; pudiendo así tener una base de datos centralizada en un servidor. La gestión externa de la base de datos se realiza a través de código.

---

<sup>5</sup> SQL: Structure Query Language, lenguaje de consulta estructurada.

El único inconveniente con respecto a las demás herramientas es que no dispone de formularios para la gestión sencilla de los datos. En el caso de "Access" o "Base", la parte de código es semitransparente para el usuario. Pero en MySQL no lo es, necesita una aplicación detrás para que éste sea transparente. Por este motivo, se decidió crear una aplicación visual mediante "Visual Studio", para el acceso, control y manipulación de la información almacenada en la base de datos. Para el desarrollo de la base de datos, se utiliza la herramienta MySQL Workbench. Esta herramienta permite desarrollar gráficamente una base de datos (tablas y relaciones) de forma gráfica.

### 3.2.3. NI MATE 2

Es un programa que se encarga de recibir y gestionar los datos que capta el dispositivo Kinect V2. Gracias a un plug-in dedicado para Blender, se pueden realizar animaciones y aplicaciones con esta herramienta 3D. La versión gratuita es la utilizada y puede ser utilizado también para otros tipos de sensores como Leap Motion, PrimeSense Carmine, Intel RealSense, OpenNI y OpenNI2. Con esto, es suficiente para el trabajo que se va a realizar con la aplicación a desarrollar. Con la versión de pago, se pueden detectar hasta 6 usuarios distintos, además de añadir más sensores para ser gestionados a la vez y otras funciones más<sup>6</sup>. En el anexo 1, se explica cómo configurar este programa para el funcionamiento de la aplicación desarrollada.

### 3.2.4. BLENDER

En el ordenador en el que se ejecutará la aplicación final desarrollada no es necesario instalar Blender, pero es necesario para el desarrollo. Se trata de un programa gratuito multi-plataforma dedicado al modelado, animación, creación de gráficos y videos, además se puede realizar aplicaciones tridimensionales. En la mayoría de los casos no requiere de conocimientos del lenguaje de programación Python, pero para la aplicación a desarrollar son necesarios. Es una herramienta freeware y de código abierto donde usuarios expertos realizan grandes aportaciones para su mejora.

### 3.2.5. VISUAL STUDIO

Una herramienta potente de programación de Microsoft. En ella, se pueden crear aplicaciones de todo tipo. Dispone de las librerías típicas de Windows y otras que se pueden agregar como en este caso la librería de MySQL. La aplicación se realizará a través de Visual C# (lenguaje de programación en C#).

### 3.2.6. FUNCIONAMIENTO DE LOS DISTINTOS PROGRAMAS

1. Para poderse comunicar con un dispositivo externo que, por lo general, no suele ser común en la configuración de un ordenador personal, se necesita de unos drivers para su uso correcto. En el caso del dispositivo Kinect V2, se instalan los drivers a la vez que se instala el SDK<sup>7</sup>. Por ello, se debe instalar la versión más reciente, el SDK 2.0 de Kinect para Windows. En él, además de todo lo necesario para el funcionamiento del dispositivo, se incluyen otras aplicaciones para la comprobación de los distintos sensores, aplicaciones de ejemplo y código de programación para entender su funcionamiento. Con este SDK se puede acceder a las librerías para poder controlar el dispositivo Kinect V2 a gusto del usuario. En este caso, en vez de investigar y desarrollar una herramienta personalizada, se decidió utilizar la aplicación Ni Mate 2, gratuita que realiza este mismo paso.
2. Una vez instalado los drivers del dispositivo Kinect V2, Ni Mate 2 hará el resto. Este programa realiza la conexión con este periférico, recibe las articulaciones del

---

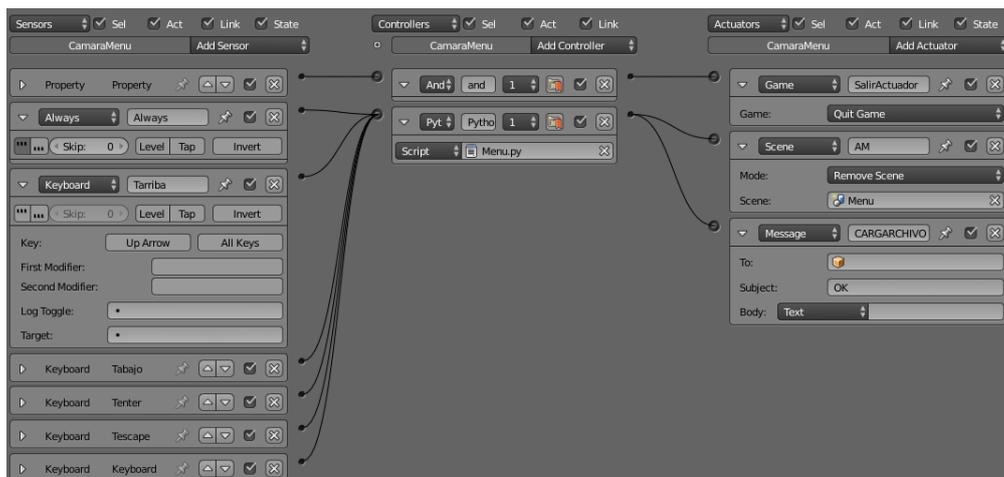
<sup>6</sup> <https://ni-mate.com>

<sup>7</sup> <https://developer.microsoft.com/en-us/windows/kinect/tools>

usuario que se encuentre delante del dispositivo Kinect V2 y los retransmite hacia el plug-in instalado en Blender.

3. Con el plug-in de Ni Mate instalado, y recibiendo la información captada por el dispositivo Kinect V2, se puede utilizar para realizar la aplicación a desarrollar.

Blender dispone de varios motores gráficos, para el renderizado: el Interno, llamado “Blender Render”, que es el motor gráfico original desde que existe esta herramienta; Cycles, que utiliza la técnica de trazado de rayos de luz y crea imágenes mucho más realistas; y, por último, “Blender Game”, que convierte a Blender en un videojuego. Este último es el que se ha utilizado para la aplicación del proyecto. Con este motor gráfico, se puede crear una aplicación ejecutable. Dentro del motor de renderizado “Blender Game” se puede configurar cada objeto de la escena individualmente de forma sencilla. A esto se le llama “Logic Bricks”, puede disponer de un número ilimitado de sensores, controladores y actuadores. Los sensores se conectan a los controladores y estos a los actuadores. En ningún caso se puede conectar un sensor a un actuador sin pasar por un controlador. Los sensores son todo tipo de elemento que introduce una señal o datos a los controladores. Pueden ser: el movimiento del ratón, pulsar una tecla del teclado, control de un joystick, etc... además de sensores virtuales como: un reloj, sensor de proximidad, sensor de choque, un retardo, recepción de un mensaje y el más importante una propiedad. Los actuadores son puertas lógicas y el más importante un script con lenguaje Python. Por último, los actuadores pueden ser una acción, un movimiento, cambiar el valor de una propiedad, enviar un mensaje, parentesco con otro objeto, editar el objeto, visibilidad, etc...



**IMAGEN 13: EJEMPLO DE FUNCIONES QUE SE PUEDEN ATRIBUIR A UN OBJETO. IZQUIERDA LOS SENSORES, CENTRO LOS CONTROLADORES Y DERECHA LOS ACTUADORES.**

Es muy importante el poder leer y escribir propiedades, porque este motor de renderizado tiene una peculiaridad. Cada frame que procesa, ejecuta el código como si fuese la primera vez, por tanto, si se inicializa una variable con un valor, al próximo frame se debe volver a inicializar. Los valores de las propiedades de los objetos, en cambio, no varían con el tiempo; por ello, para memorizar datos importantes, se utilizarán las propiedades de los objetos. El programa debe funcionar a 60 frames por segundo o como es lo mismo a 60 fps. Como la carga de trabajo es insignificante, se convertirá en el valor normal de trabajo.

Para que la aplicación 3D pueda acceder a la base de datos, se requiere instalar previamente Connector/Python de MySQL. Con este conector, Python dispondrá de la librería necesaria para realizar la conexión y consultas al servidor.

Blender utiliza dos sistemas de medidas angulares: Euler y cuaternio.

- El sistema angular de Euler necesita 3 valores para realizar una rotación del vector. Rotando desde el eje X, eje Y, y eje Z. Pero hay un problema y son las prioridades de ejecución. Porque si un vector localizado en el eje Y realiza una pequeña rotación por el eje X y después con el eje Y realiza otra rotación, dependiendo de las prioridades ese vector rotará a sí mismo (si la prioridad es la rotación del eje Y) o rotará respecto a su nueva posición (en el caso de ser el eje X el prioritario). Este es el problema del cálculo angular de Euler llamado "bloqueo del cardán"<sup>8</sup>. Dispone de 6 modos de prioridades para las tres rotaciones.
  - Por el contrario, cuaternio es un sistema numérico que extiende los números complejos. Generalmente se utilizan para modelizar y calcular rotaciones al espacio debido a sus propiedades geométricas. En este caso, en un entorno de tres dimensiones necesita cuatro campos de datos, los ejes X, Y, Z (números complejos) y el cuarto que es W (número real).
4. MySQL es un sistema de gestión de bases de datos desarrollado por Oracle Corp. Una de las versiones que dispone MySQL es la Community, la cual es gratuita (además de disponer de la ayuda de toda la comunidad para las dudas) y para la poca complejidad del programa a desarrollar, no es necesaria otra versión. MySQL crea un servidor que puede ser accedido desde cualquier dispositivo conectado a Internet, y éste gestiona la base de datos, consultas externas y demás funciones. Para el correcto funcionamiento de la base de datos, es necesario tener instalado el servidor y, opcionalmente, se pueden instalar otros programas que ayudan a la gestión de la base de datos: Notifier, para un control sencillo del servidor; Connector que, dependiendo del lenguaje de programación de las aplicaciones que vayan a acceder al servidor, se instala uno o varios Connector's diferentes; Workbench, para la programación desde cero de una nueva base de datos. Este último es el utilizado para la creación de las tablas y relaciones de la base de datos de la aplicación de este proyecto. Una base de datos es, simplemente, una selección de tablas de datos (con filas y columnas de información), las cuales están interrelacionadas entre ellas para abarcar mayores prestaciones en la gestión de datos. MySQL utiliza el lenguaje de programación SQL. Para realizar consultas, introducir datos, crear nuevas tablas, se debe utilizar este lenguaje. Para el ordenador que ejecutará la aplicación final desarrollada, sólo será necesario tener instalado el servidor y los conectores, mientras que para el desarrollo de la aplicación se han utilizados todos los programas citados.
5. Ésta base de datos no podría existir sin una interfaz gráfica de usuario (GUI). Esta herramienta sólo será utilizada para la creación de la aplicación final que será la utilizada por el fisioterapeuta y por el paciente. Para ello, se va a utilizar Visual Studio, donde se creará el entorno visual que podrá acceder a los datos de la base de datos y ejecutar el entorno 3D donde el paciente realizará sus ejercicios. Visual Studio<sup>9</sup> de Microsoft (y toda la comunidad detrás de todos los tutoriales y ayudas en la red), se ha creado un entorno sencillo y fácil para el usuario. Visual Studio está preparado para la programación de diversos lenguajes, y, dependiendo del lenguaje a utilizar, dispone de diferentes modos donde se pueden crear Forms (Formularios visuales) de una forma gráfica. Es mucho más sencillo que si se tuviese que realizar todo mediante código de programación. Se optó por realizar la aplicación con Visual C# con lenguaje C# que dispone de Forms sencillos de diseñar, y que, mediante el Connector/Net de MySQL, permite acceder al control del servidor que gestiona la base de datos.

---

<sup>8</sup> Bloqueo del cardán consiste en la pérdida de grado de libertad que ocurre cuando en los ejes, dos de los tres rotores se colocan en paralelo, bloqueando al otro rotor. Web: [wikipedia.org](http://wikipedia.org)

<sup>9</sup> <https://www.visualstudio.com/es/>

---

# CAPÍTULO 4. DESARROLLO DE LA APLICACIÓN

La aplicación finalmente desarrollada integra cuatro partes importantes: el dispositivo Kinect V2, la base de datos, el entorno visual (GUI) y de interfaz con la base de datos MySQL, y el entorno 3D con los ejercicios a realizar por el paciente.

Para ello, es necesario instalar un servidor de base de datos MySQL, instalar unos conectores específicos para poder acceder a la base de datos, instalar y configurar las aplicaciones para el funcionamiento del dispositivo Kinect V2, y crear los archivos ejecutables para, por un lado, acceder a la base de datos mediante formularios, y un entorno virtual tridimensional para la ejecución de los ejercicios programados.

La aplicación final integra dos programas, uno principal realizado en Visual C# que controla la interfaz de gestión y de usuario, y otro, realizado en Blender, para el entorno gráfico en 3D para la ejecución de los ejercicios controlados por dispositivo Kinect V2. El segundo será llamado por el primero cada vez que un usuario inicie una sesión de ejercicios.

## 4.1. INSTALACIÓN Y CONFIGURACIÓN DEL DISPOSITIVO KINECT V2

En primer lugar, se pretende conectar, instalar y comprobar el correcto funcionamiento del dispositivo de captura de movimientos en el ordenador preparado para éste fin; en segundo lugar, averiguar cómo realizar el tratamiento de los datos y conseguir el buen funcionamiento de la aplicación final.

Para poder controlar el dispositivo Kinect V2, hace falta instalar unos drivers para que el ordenador reconozca este dispositivo y puedan comunicarse. Estos drivers solo se encuentran en el SDK que Microsoft ofrece para el dispositivo Kinect V2.<sup>10</sup>

Una vez instalado, el propio SDK dispone de una aplicación llamada “Configuration Verifer” que analiza el correcto funcionamiento del dispositivo Kinect V2 en el equipo que se conectó.



IMAGEN 14: PANTALLA DE KINECT V2 CONFIGURATION VERIFIER, INDICANDO EL CORRECTO FUNCIONAMIENTO (SALVO UN PAR DE ADVERTENCIAS) DEL DISPOSITIVO KINECT V2 CON EL ORDENADOR INSTALADO.

---

<sup>10</sup> Web de descarga del SDK de Kinect: <https://developer.microsoft.com/es-es/windows/kinect/tools>

En muchos casos está reportado por usuarios en diversos foros, aparece una advertencia en la conexión del USB y en los datos recibidos de las cámaras de profundidad y de color. Esto se debe a que, en muchos ordenadores con puertos USB 3.0, o el hardware no llega a cumplir al 100% del estándar 3.0 o los drivers instalados por Windows, al ser genéricos, no consiguen sacar todo el rendimiento que deberían. De todas formas, para este caso no implica problema alguno para el correcto funcionamiento de esta aplicación.

## 4.2. INSTALACIÓN DE LA APLICACIÓN NI-MATE 2

La instalación de Ni Mate 2 es sencilla. Una vez instalada, y abrir la aplicación, se observa una sencilla ventana, donde en este caso solo se necesita disponer de la ubicación del esqueleto. Para ello, se marcará la opción “Skeleton tracking” dentro del apartado “Kinect for Windows” en la columna de la izquierda, tal como se aprecia en la imagen 15.

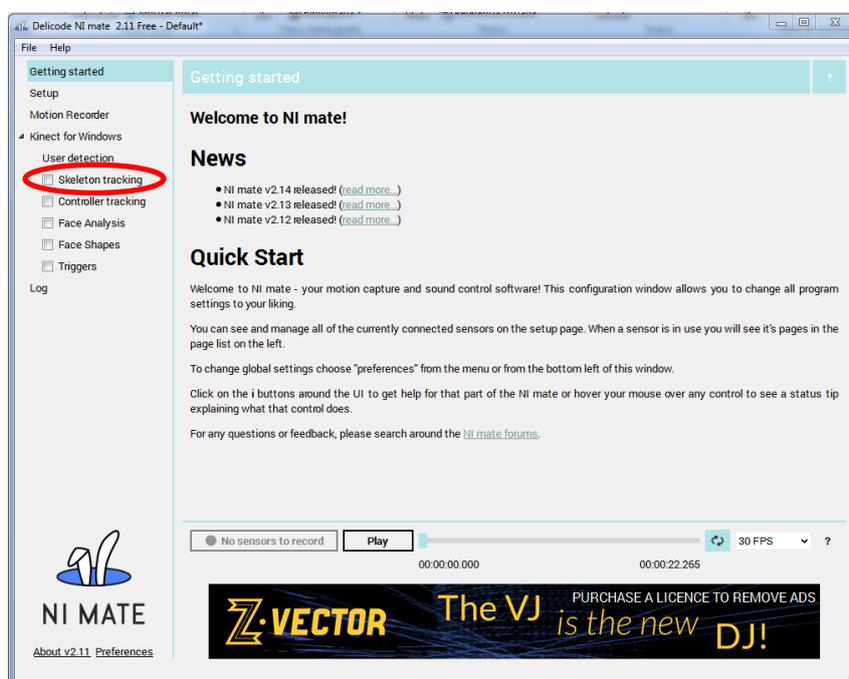


IMAGEN 15: CONFIGURACIÓN DE NI-MATE.

Con esto, la aplicación retransmitirá la posición de cada articulación que el dispositivo Kinect V2 detecte del usuario.

El funcionamiento de este programa es el de retransmitir mediante TCP/IP, por defecto, a la dirección 127.0.0.1:7000 (es decir, al propio equipo) los datos recibidos del dispositivo Kinect V2. El plug-in instalado en Blender capta esa información para utilizarla para su propio beneficio.

## 4.3. INSTALACIÓN Y CONFIGURACIÓN DEL PLUG-IN FOR NI-MATE DE BLENDER

Blender permite instalar y habilitar a parte de todas sus herramientas, externas almacenadas en un archivo “.py”. La instalación del plug-in dentro de Blender está explicada en el anexo 6.

Con el plug-in instalado, Blender puede recibir la información que obtiene Ni-Mate del dispositivo Kinect V2.

## 4.4. DESARROLLO DE LA APLICACIÓN TRIDIMENSIONAL EN BLENDER

El plug-in de Ni Mate 2 instalado dentro de la herramienta Blender crea “Empties”, que representan cada uno de ellos, una articulación que reconocida por el dispositivo Kinect V2 y los ubica en el espacio tridimensional. Cada “Empty” tiene asignado un nombre específico que representa una articulación. Como se muestra en la figura, se puede ver la representación de una persona con los brazos levantados.

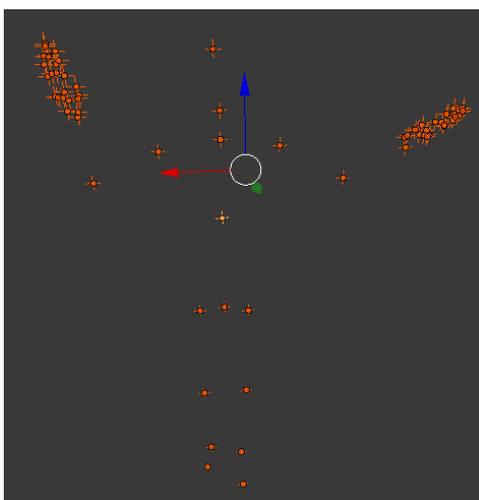


IMAGEN 16: REPRESENTACIÓN DE LA UBICACIÓN DE LOS “EMPTIES” DE UNA PERSONA CON LOS BRAZOS LEVANTADOS.

### 4.4.1. AVATARES UTILIZADOS

Un avatar es la representación virtual que hace referencia a una persona. Esta representación será una malla con una cantidad de vértices en el espacio tridimensional virtual que de por sí, no puede realizar ningún tipo de movimiento. Para ello, se asigna un elemento no visual dinámico que pueda realizar funciones por él. Para la malla, se utiliza un objeto tipo esqueleto que contiene huesos, para que el movimiento sea el más real posible. Blender dispone de un esqueleto predeterminado que puede incluir hasta los complejos y precisos movimientos de la cara.

En primer lugar, el objetivo a conseguir es el de manejar un avatar 3D con el dispositivo Kinect V2. Para ello, se tiene que crear u obtener un avatar. Hay muchas webs donde se comparten, de forma gratuita, entre usuarios de modelismos 3D, todo tipo de recreaciones tridimensionales. Uno de ellos es [free3d.com](https://free3d.com) (enlace del avatar femenino<sup>11</sup>, enlace del avatar masculino<sup>12</sup>). Esta web es donde se obtienen dos recreaciones humanas (masculina y femenina) tridimensionales con sus texturas (colorido que se añade a un objeto 3D para que su recreación sea más realista), para las primeras pruebas del dispositivo Kinect V2 con Blender. En muchos casos, los usuarios de este tipo de webs, no utilizan la misma herramienta para crear sus objetos 3D y, por tanto, esos archivos se pueden guardar en formatos no soportados por la herramienta utilizada. De todas formas, el propio Blender dispone de distintos formatos soportados para ser importados.

<sup>11</sup> Avatar femenino: <https://free3d.com/3d-model/sport-woman-47016.html>

<sup>12</sup> Avatar masculino: <https://free3d.com/3d-model/sport-man-64993.html>



IMAGEN 17: IZQUIERDA: AVATAR FEMENINO, DERECHA: AVATAR MASCULINO.

Estas mallas tenían una peculiaridad especial, estaban divididas por elementos: cara, cabeza, pelo, brazos, camiseta, pantalones y zapatillas. Para poder ser utilizados se tienen que unir todas las mallas en una sola, para que la asignación a los huesos sea la correcta. Se tuvo que unir cada elemento, vértice a vértice, para que no quedase ningún vértice suelto y pudiese realizar deformaciones inapropiadas.

En primer lugar, se tuvo que unir los distintos objetos en uno solo. Para ello se deben seleccionar todos los objetos, y pulsar las teclas Control + J (Join) para unir las mallas en un objeto.



IMAGEN 18: LISTA DE OBJETOS DE UN AVATAR

Una vez unidos en un objeto, se pasa a modo edición para comprobar las uniones entre distintas mallas que se encuentran por separado.

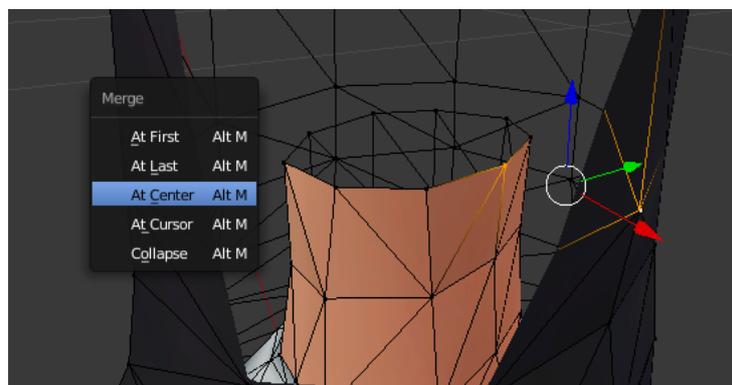
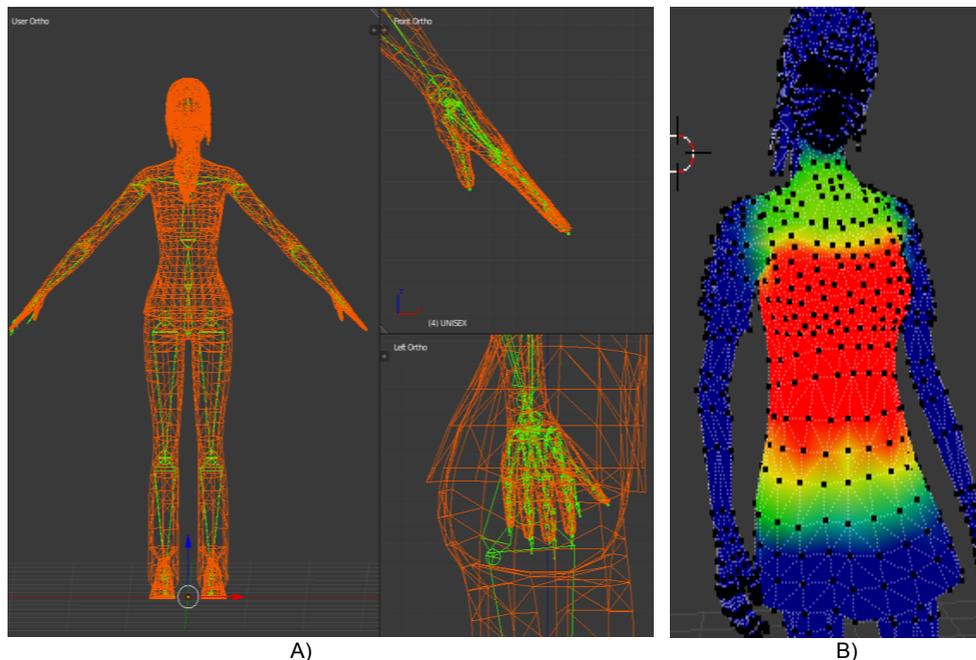


IMAGEN 19: UNIÓN DE DOS VERTICES DE DOS MALLAS SEPARADAS.

Yendo de vértice en vértice, realizar unión entre mallas. En muchos casos los vértices extremos de una malla no coinciden con los de otra malla. Para ello seleccionando un vértice final de una malla con el vértice más próximo de la otra malla. Por ejemplo como se puede observar en la imagen 19, se selecciona el vértice del final de la malla del calzado y el vértice más próximo de la malla pantalón. Pulsando las teclas Alt + M

(Merge) y elegir la opción At Last, une el primer vértice con el segundo en la ubicación del segundo. Este proceso se debe de repetir por todos los vértices por unión y en todas las uniones de mallas.

Una vez se tiene el objeto a utilizar, es hora de asignarle un esqueleto que se encargue de deformar la malla, simulando la forma natural de los movimientos humanos. Un esqueleto virtual de Blender dispone de una cantidad de huesos los cuales están unidos entre sí. Siempre hay un hueso principal (hueso padre), que es donde los demás huesos se van uniendo hasta crear una red de huesos al gusto del usuario. Blender puede asociar de forma automática cada hueso del esqueleto a cada parte de la malla por proximidad al hueso, pero la figura 3D debe estar bien diseñada y los huesos colocados en el interior de la mejor forma, para que no haya problemas. Primeramente, se debe de colocar todo el esqueleto dentro del personaje y recoloca en algunos casos para que coincida cada hueso con el tamaño y la posición de la figura. La función de asociación, en este caso trata de asociar vértices de la malla a un hueso específico, de tal forma que el movimiento que realice dicho hueso, los vértices asociados también los realizarán. Además, se pueden asociar gradualmente varios huesos al mismo vértice para que la deformación de la malla sea más ecuánime a la realidad.



**IMAGEN 20: A) COLOCACIÓN DE LOS HUESOS (COLOR VERDE) DENTRO LA MALLA (COLOR NARANJA) PROCURANDO QUE LAS ARTICULACIONES COINCIDAN. B) ASOCIACIÓN DE VÉRTICES A UN HUESO, EN ESTE CASO ESTÁN ASOCIADOS AL HUESO CENTRAL DE LA COLUMNA VERTEBRAL QUE EN SU**

En la imagen anterior, se pueden ver cómo afectará el movimiento del hueso con los vértices asociados. Lo representado en color rojo realizarán tal cual los movimientos del hueso, y gradualmente los otros vértices de distinto color realizarán menor movimiento hasta llegar a las partes representadas en color azul, donde dichos vértices no se moverán por el efecto del hueso en cuestión. En muchos casos, la asociación automática de Blender no realiza correctamente la asociación de los vértices con los huesos. Se puede ver que eso sucede cuando el hueso en cuestión o los contiguos, al realizar un movimiento, la figura se deforma de una forma no correcta. Por ello, se debe comprobar hueso por hueso que las asociaciones son correctas y sus movimientos simulen a los fisiológicos que puede realizar una persona. Una vez que la figura está bien asociada con los huesos es momento de que empiece a moverse.

Un esqueleto tiene una estructura tipo árbol donde un hueso es el principal, y los demás derivan de él.

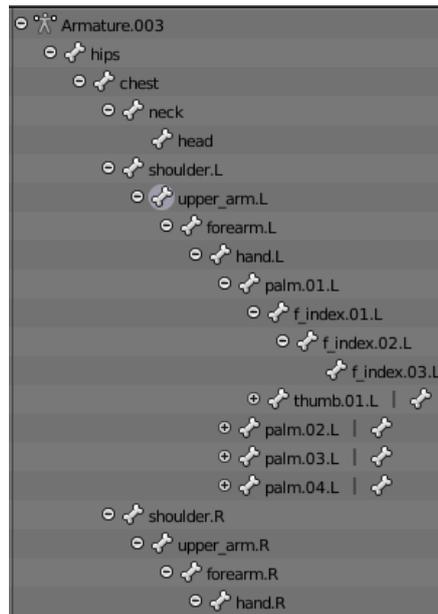


IMAGEN 21: ESTRUCTURA DE LOS HUESOS DE UN ESQUELETO.

Por ejemplo el hueso que representa el cuello, está conectado a su principal que es el hueso de la columna vertebral, mientras que del hueso del cuello, deriva el hueso que representa a la cabeza, siendo éste una articulación final. Todos los huesos tienen un principio (cabeza) y un final (cola), y cuando se le aplica una rotación, el punto que no variará su posición es la “cabeza” que a su vez es la ubicación de la cola del hueso anterior, mientras que la “cola” cambiará su ubicación al aplicar la rotación, manteniendo siempre el tamaño del hueso.

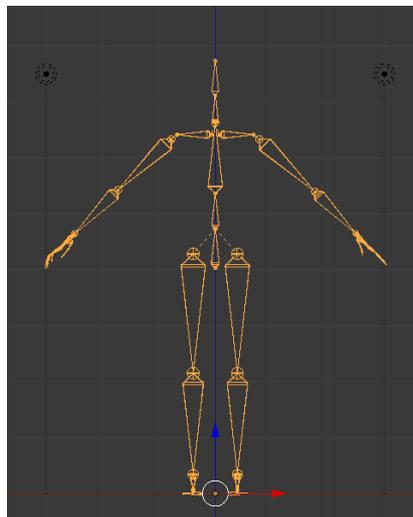


IMAGEN 22: ESQUELETO EN POSICIÓN NEUTRA.

#### 4.4.2. CREACIÓN DEL ENTORNO 3D

Para la creación del entorno 3D se ha creado un cubo con una textura de color ocre claro y unas lámparas para iluminar el interior. A las texturas de los avatares, se les cambia la propiedad de emisión de luz a un 100% para que el motor de renderizado no tenga que procesar los efectos de las sombras.

### 4.4.3. MOTOR DE RENDERIZADO

En primer lugar, se realizaron pruebas con el motor de renderizado “Blender Render”, pero para poder disponer de una aplicación ejecutable requiere del motor de renderizado “Blender Game”. Este motor se activa pulsando la letra “P” del teclado, y entra en modo ejecución. Con “Blender Game” se puede exportar a un archivo ejecutable, el cual, permite no depender de Blender y ejecutarse en cualquier ordenador personal con SO Windows.

Las diferencias entre estos dos tipos de motores de renderizado son que, entrando en el código de programación, “Blender Render” utiliza la librería “bpy”, y todas las funciones de Blender residen en dicha librería; mientras que “Blender Game” utiliza la librería “bge”. Para poder crear el ejecutable se tiene que utilizar dicha librería. Probando con “Blender Game” el avatar no realiza ningún movimiento. En primer lugar, se debe activar el esqueleto para que se asocie de nuevo con la malla del avatar y para que pueda moverse. La función que dispone Blender para copiar la ubicación de un objeto a otro, en “bge” no existe implícitamente. Además la función seguidor que es el utilizado para que los huesos apunten donde se encuentra el “Empty” que representa la articulación, no existe. Es por todo esto que se opta por empezar a crear código ya que las opciones que dispone en las “Logic Bricks”, el motor “Blender Game” son muy limitadas mientras que utilizando código se pueden acceder a otras funciones no accesibles.

La web de Blender<sup>13</sup> dispone de una sección donde se encuentra un catálogo con toda la información necesaria para realizar scripts con Python, ya que la librería “bpy” y “bge” dentro de Python son exclusivas de Blender. Con la combinación de las funciones “`print(dir(“Nombre de la función”))`” que imprime por consola las funciones del elemento a introducir dentro de los paréntesis, se ve las posibles funciones a utilizar. Para más información se busca en la web anteriormente descrita para comprender el funcionamiento. Es entonces cuando se decide utilizar tutoriales en Youtube y otras plataformas que expliquen cómo realizar los scripts y entender mejor todo el proceso de crear una aplicación 3D a través de Blender. Para ello, el canal público de Miguel Roque en Youtube [5-ROQUE], dispone de varias listas de reproducción con contenido específico de Blender.

Estos vídeos y otros encontrados en dicho canal, además de diferente documentación, han servido para el desarrollo de la aplicación 3D.

Una de las pruebas que se realizó fue asociar mediante código, el “Empty” de una articulación, que es un objeto no visible, con un elemento que sí fuera visible. Se pudo comprobar que el elemento visible se desplazaba por el espacio, según como se moviese el usuario.

La siguiente prueba consistía en asociar el esqueleto completo a un único “Empty”. Cómo la malla del avatar ya está asociada al esqueleto, se comprueba si reacciona. El resultado fue negativo. Para solucionar esto, se debe activar el actuador específico del propio esqueleto. Es entonces cuando se desplaza por el espacio todo en un bloque.

Otra prueba es asociar un hueso del esqueleto a un solo “Empty” para comprobar si se desplaza. Pero en este caso, la asociación no puede ser realizada, porque con este motor de renderizado, sólo se permite asociar objetos, y un hueso es una parte de un objeto. El código que se permite utilizar para el tratamiento de un hueso de un esqueleto es bastante limitado. En primer lugar, cambiar el valor de la ubicación de un hueso sólo es permitido si dicho hueso no está unido a ningún hueso padre, siendo el hueso principal el único que no tiene parentesco. Hay dos funciones “pose\_head” y

---

<sup>13</sup> <https://docs.blender.org/>

“pose\_tail” que son los que contiene la ubicación de la cabeza del hueso y su cola; pero estas dos funciones son de “solo lectura” y no permiten ser modificadas. Las únicas que podrían permitir realizar cambios son las funciones de rotación. Esto implica que el punto principal que en muchos casos está unido al hueso padre, es el punto fijo donde se desplazará el hueso dependiendo del ángulo que se le aplique.

Dispone de 2 formas de ejecutar rotaciones: rotación conjunta y rotación respecto al hueso padre. Para esto, hay que tener conocimientos vectoriales.

- Para entender la rotación conjunta primero hay que entender que un hueso, cuando es creado, se puede tratar como un vector donde tiene un punto inicial y otro final en el espacio tridimensional. La rotación conjunta, utiliza ese vector que es el hueso como si fuese un eje de coordenadas, para ser exactos el eje de coordenadas Y, y las rotaciones que se apliquen al hueso serán respecto a esas coordenadas. De esta forma crea sus propias coordenadas, o como en Blender se conoce, coordenadas locales (explicado en la siguiente página). Por tanto, dependiendo donde esté apuntando el vector y una variable llamada “roll”, se determinará la dirección de los ejes X, Y, Z. Los ejes de coordenadas X y Z se sitúan perpendicularmente a Y, y dependerán de una variable específica llamada “roll” que determina hacia donde apuntan estos ejes.
- Para el caso de rotación respecto al hueso padre hace la misma función. En este caso, el vector que identifica el eje Y es el hueso padre del hueso a rotar. De igual forma X y Z dependerán de “roll”.

Mediante un conversor de ángulos de Euler a cuaternio, o con un constructor de objetos, se puede conseguir los datos necesarios e indicarle al hueso en cuestión que realice dicha rotación. Para crear un cuaternio desde cero se necesita los ángulos de rotación visto desde el eje X, Y, Z y los grados totales entre dos vectores.

Para conocer la rotación a aplicar, se obtiene el vector inicial, que es la ubicación inicial de la cabeza del hueso en cuestión y de su cola. El vector final, que será la ubicación de los “Empties” de las articulaciones que representaría donde se encuentra el supuesto hueso que debería de localizarse entre estos dos objetos. Con esto bien calculado se podría conseguir el movimiento de todos los huesos del avatar.

Blender dispone de dos tipos de coordenadas, el propio de cada objeto llamado “Local” y las coordenadas generales llamadas “Global”. Los “Empties” que representan las articulaciones, las coordenadas globales son las que varían según los datos enviados por el dispositivo Kinect V2 y su ubicación local es siempre la misma. En el caso de un esqueleto y de sus huesos, el esqueleto al ser el objeto, éste dispone y le afectan las coordenadas globales, mientras que cada hueso, únicamente dispone de coordenadas locales e individuales para cada hueso. Y por ello, las coordenadas globales son únicamente informativas (de solo lectura). Esto provoca el siguiente inconveniente: para calcular el vector entre dos “Empties” se hace a través de los ejes de coordenadas globales, mientras que cada hueso del esqueleto tiene su propio eje de coordenadas que no coinciden en ningún caso con las globales.

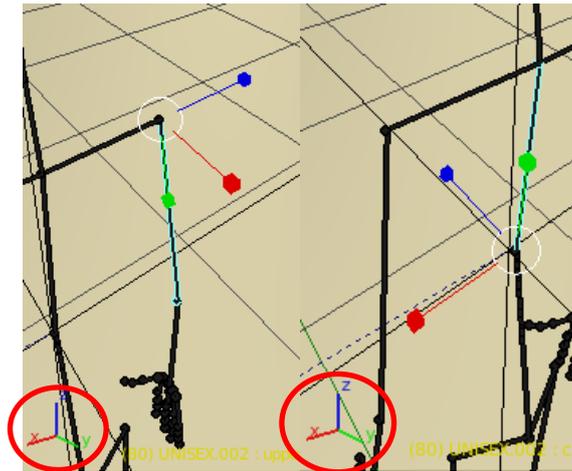


IMAGEN 23: DIFERENCIAS ENTRE COORDENADAS LOCALES Y COORDENADAS GLOBALES.

Como se muestra en la imagen, a la izquierda en pequeño las coordenadas globales. En la imagen de la izquierda, las coordenadas locales del brazo. En la imagen de la derecha, las coordenadas locales de la columna entre el torso y el inicio del cuello. Con esta imagen se puede ver la dificultad que existe para compensar la diferencia de coordenadas con las coordenadas globales que se rigen los demás objetos de la escena.

Este problema no se puede resolver con facilidad, y para eso, se decidió cambiar el planteamiento. Por ello, la solución que se ha implementado ha sido la de crear animaciones del avatar realizando el movimiento en cuestión y sobrepasando un poco los límites humanos. Las animaciones, finalmente, son una recreación de varios frames (imágenes fijas) de un movimiento, y, con ello, se crea la ilusión del movimiento realizado. Una de las funciones que permite el motor de renderizado “Blender Game” es que a través de una propiedad de tipo entera almacenada en el objeto tridimensional, se puede indicar qué frame se desea que aparezca en pantalla. Realizando un cálculo sencillo, se puede saber qué ángulo está realizando el paciente y hacia qué dirección, y con ello, mediante una regla de tres, indicar qué frame es donde está el brazo del paciente. De este modo, se ha evitado el tener que averiguar la rotación de todos los huesos e introducirlos al esqueleto del avatar.

#### 4.4.4. OBTENCIÓN ÁNGULOS A PARTIR DE LOS DATOS CAPTURADOS POR EL DISPOSITIVO KINECT V2

Los “Empties” de las articulaciones que se mueven a través del usuario, se ubican en un espacio tridimensional. Como ya se ha explicado, cada “Empty” es una articulación reconocida por el dispositivo Kinect V2 y transmitida a Blender en coordenadas X, Y, Z. Con ello, se pueden hacer cálculos de los vectores que simulará a un hueso. La diferencia respecto al método utilizado para calcular la diferencia entre los ejes de coordenadas locales con los globales, es que los ángulos que se calculan son vectores representados en los ejes de coordenadas globales.

Sabiendo cómo es el vector inicial y cuál es la posición del paciente a través de los “Empties”, se obtiene el ángulo entre los dos vectores. En el caso del brazo, los cálculos son los siguientes:

Se toma como referencia estas articulaciones que transmite el dispositivo Kinect V2: hombro y codo. Cada articulación tiene sus ejes de coordenadas X, Y, Z. [6-NUÑEZ]

##### 4.4.4.1. ABDUCCIÓN

El ejercicio abducción glenohumeral es realizar el movimiento levantando el brazo desde posición de descanso (apuntando la mano hacia el suelo) lateralmente. Para

calcular la diferencia entre dos vectores requiere disponer de un vector fijo para realizar el cálculo del ángulo del vector variable. Para ello, se utiliza como vector fijo la linealidad entre el punto del cuello y el punto del torso. Estos cálculos son los siguientes.

$$V1[X, Y, Z] = Torso[X, Y, Z] - Cuello[X, Y, Z]$$

$$V2[X, Y, Z] = Codo[X, Y, Z] - Hombro[X, Y, Z]$$

$$Ang = \cos^{-1} \left( \frac{V1[X] \cdot V2[X] + V1[Z] \cdot V2[Z]}{\sqrt{V1[X]^2 + V1[Z]^2} \cdot \sqrt{V2[X]^2 + V2[Z]^2}} \right)$$

$$Ang = \left( \frac{W \cdot 180}{\pi} \right)$$

**ECUACIÓN 1: FORMULACIÓN PARA EL CÁLCULO DEL MOVIMIENTO ABDUCCIÓN GLENOHUMERAL.**

El defecto que tiene el dispositivo Kinect V2, es que no detecta correctamente la ubicación de una articulación, y, dependiendo de la posición en la que se encuentre el usuario, varía la posición de la articulación.

#### 4.4.4.2. FLEXIÓN

El ejercicio flexión glenohumeral es realizar el movimiento levantando el brazo hacia delante y hacia atrás. En este caso, el movimiento total supera los 180° con el inconveniente que el resultado de la fórmula es entre 0° y 180°. Estos cálculos son los siguientes.

$$V1[X, Y, Z] = Codo[X, Y, Z] - Hombro[X, Y, Z]$$

$$V2[X, Y, Z] = Torso[X, Y, Z] - Cuello[X, Y, Z]$$

$$Ang = \cos^{-1} \left( \frac{V1[Y] \cdot V2[Y] + V1[Z] \cdot V2[Z]}{\sqrt{V1[Y]^2 + V1[Z]^2} \cdot \sqrt{V2[Y]^2 + V2[Z]^2}} \right)$$

$$Ang = \left( \frac{W \cdot 180}{\pi} \right)$$

**ECUACIÓN 2: CÁLCULO DEL MOVIMIENTO FLEXIÓN GLENOHUMERAL.**

V2 es el vector fijo mientras que V1 es el vector variable. Al necesitar de un vector fijo para realizar el cálculo, se decidió utilizar una parte del cuerpo que se encuentre en una posición vertical obligada para la correcta ejecución del ejercicio. Con esto, aunque el paciente se inclinase o acostase, el ángulo entre los dos vectores seguiría siendo el mismo. Con este cálculo el ángulo va de 0° a 180°; por tanto de la posición de reposo hasta el brazo en posición vertical. Ya sea si el movimiento es hacia delante o hacia atrás, el valor es positivo. Por tanto, conociendo la ubicación de las articulaciones, se establece una condición, que cuando el brazo este adelantado al hombro, el ángulo sea positivo, mientras que si es a la inversa, el ángulo es negativo.

#### 4.4.4.3. ROTACIÓN

La rotación glenohumeral consiste en tener el brazo flexionado; mover el antebrazo hacia la barriga o hacia el exterior del cuerpo, mientras el codo se mantiene pegado al cuerpo. El cálculo del ángulo viene del vector variable y del vector fijo. El vector variable es desde el codo hasta la muñeca. Mientras que el vector fijo no puede ser desde el codo hasta el hombro. Por tanto, como en el ejercicio anterior, se utilizan otras articulaciones para crear el vector. En este caso, se usa el vector desde el

hombro hasta el cuello. El cálculo del ángulo es exactamente el mismo que el anterior descrito, con siguiente diferencia:

$$V1[X, Y, Z] = \text{Cuello}[X, Y, Z] - \text{Hombro}[X, Y, Z]$$

$$V2[X, Y, Z] = \text{Muñeca}[X, Y, Z] - \text{Codo}[X, Y, Z]$$

$$\text{Ang} = \cos^{-1} \left( \frac{V1[Y] \cdot V2[Y] + V1[Z] \cdot V2[Z]}{\sqrt{V1[Y]^2 + V1[Z]^2} \cdot \sqrt{V2[Y]^2 + V2[Z]^2}} \right)$$

$$\text{Ang} = \left( \frac{W \cdot 180}{\pi} \right) - 90$$

**ECUACIÓN 3: FORMULACIÓN PARA EL CÁLCULO DEL MOVIMIENTO ROTACIÓN GLENOHUMERAL.**

#### 4.4.5. CREACIÓN DE LAS ANIMACIONES 3D

Para esta aplicación se han creado tres animaciones principales del movimiento del avatar, una por cada ejercicio base preparado. Estos ejercicios son flexión, abducción y rotación. La explicación de cómo realizar una animación en Blender está incluida en el anexo 5.

Para realizar el movimiento del ejercicio abducción glenohumeral, la articulación afectada es el hombro. En primer lugar, se va a crear la animación del movimiento. En este caso, se crean claves de rotación.

Se establece que el frame 0 corresponde con la posición del brazo en estado de reposo y que el frame 90 corresponde a la posición de 90° respecto a posición de reposo. La posición inicial de los brazos del esqueleto del avatar es aproximadamente a 45°. Como se muestra en la siguiente figura gráfica, donde la línea vertical verde a la derecha indica la posición del frame del cursor, que en este caso es el frame 90 (imagen 24 A); la línea vertical negra a la izquierda indica la posición del frame 0 (imagen 24 B); la línea roja diagonal, es la rotación del brazo en su respectiva coordenada X. Para que el movimiento visual del avatar sea lo más real posible, se asigna movimiento lineal al miembro del brazo y al hombro. La suma de los ángulos del movimiento de los miembros, es el ángulo real detectado por el dispositivo Kinect V2 respecto a la coordenada horizontal.

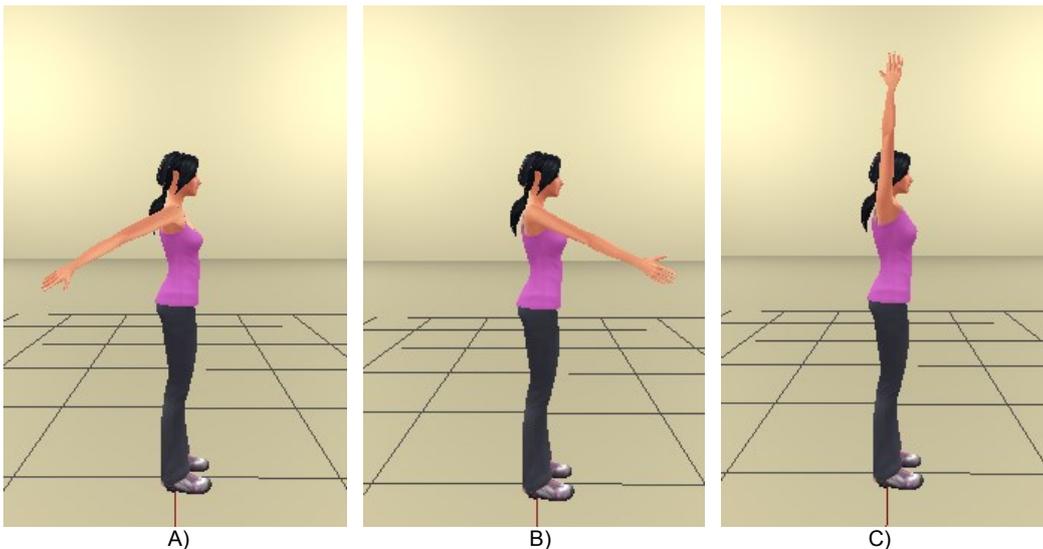
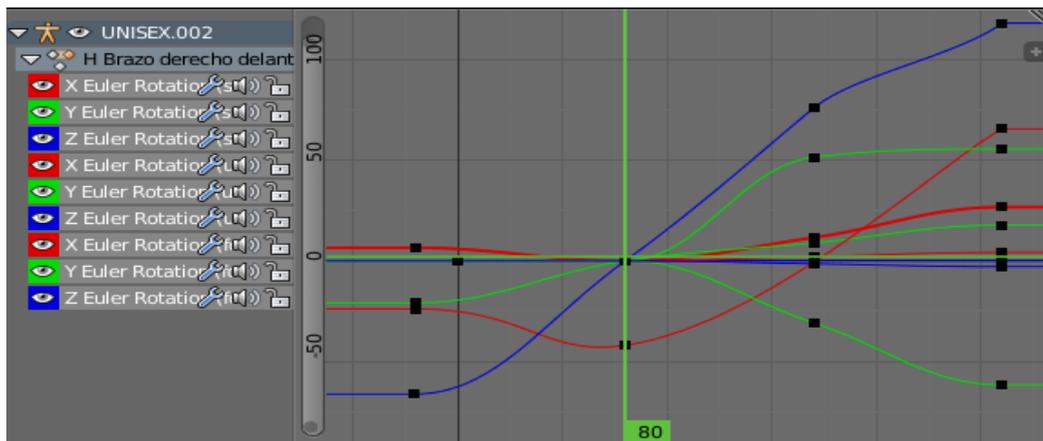
Además, se añade la posibilidad de que superados los umbrales de 0 y 90 frames continúe el movimiento. De esta forma, el brazo puede llegar a realizar el movimiento extra para pacientes que consigan superar con creces los límites establecidos.



**IMAGEN 24: GRÁFICA DE LA ANIMACIÓN DEL MOVIMIENTO FLEXIÓN GLENOHUMERAL (LATERAL). A) BRAZO EN POSICIÓN REPOSO. B) BRAZO A 90° HACIA EL LADO.**

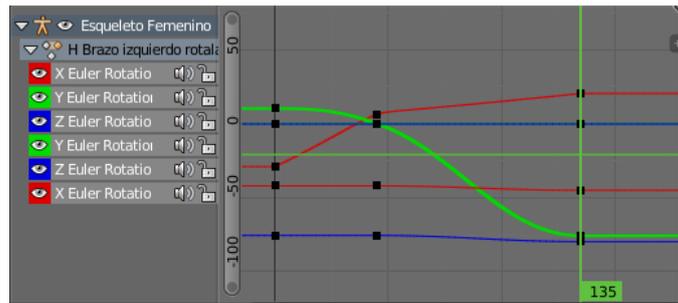
Las otras dos animaciones son ligeramente distintas a ésta.

Se establece para el ejercicio flexión glenohumeral que el frame 0 corresponde con la posición del brazo hacia atrás a  $80^\circ$  respecto al cuerpo, que el frame 80 corresponda a la posición en estado de reposo, que el frame 170 corresponde a la posición de  $90^\circ$  respecto a posición del cuerpo y por último el frame 260 corresponde a la posición de  $180^\circ$  respecto a posición del cuerpo. La posición inicial de los brazos del esqueleto del avatar es aproximadamente a  $45^\circ$  lateral por tanto la rotación no será igual de sencillo que en el ejercicio anterior. Como se muestra en la siguiente imagen, donde la línea vertical verde a la derecha indica la posición del frame del cursor, que en este caso es el frame 80 que es la posición de reposo (imagen 25 B); la línea vertical negra a la izquierda indica la posición del frame 0 que representa la posición del brazo hacia atrás a  $80^\circ$  (imagen 25 A); la línea de puntos negros a la derecha indica la posición del frame 260 que representa la posición del brazo hacia atrás a  $180^\circ$  (imagen 25 C); las líneas rojas, verdes y azules con recorrido de izquierda a derecha, representan la rotación del brazo. Para que el movimiento visual del avatar sea lo más real posible, se realizan rotaciones en los tres ejes para evitar un problema de pinzamiento de la malla que se explicará en el apartado 4.4.5.1.



**IMAGEN 25: GRÁFICA DE LA ANIMACIÓN DEL MOVIMIENTO ABDUCCIÓN GLENOHUMERAL (DELANTE-ATRÁS). A) BRAZO A 70° HACIA ATRÁS. B) BRAZO A 45° HACIA DELANTE. C) BRAZO A 180° HACIA DELANTE.**

Se establece para el ejercicio rotación glenohumeral que el frame 0 corresponde con la posición del brazo hacia el interior a 45° respecto a la linealidad frontal del cuerpo, que el frame 45 corresponda a la posición en estado de reposo y que el frame 135 corresponde a la posición de 90° respecto a linealidad frontal del cuerpo. Como se muestra en la siguiente imagen, donde la línea vertical verde a la derecha indica la posición del frame del cursor, que en este caso es el frame 135 que es la posición de a 90° (imagen 26 B); la línea vertical negra a la izquierda indica la posición del frame 0 que representa la posición del brazo a 45° (imagen 26 A); las líneas rojas, verdes y azules con recorrido de izquierda a derecha, representan la rotación del brazo.



A)



B)

**IMAGEN 26: GRÁFICA DE LA ANIMACIÓN DEL MOVIMIENTO ROTACIÓN GLENOHUMERAL. A) BRAZO A 45° HACIA EL INTERIOR. B) BRAZO A 45° HACIA EL EXTERIOR.**

#### 4.4.5.1. PROBLEMAS DE PINZAMIENTO DE LA MALLA

Otro de los problemas a solucionar es el mostrado en la siguiente imagen:



**IMAGEN 27: PINZAMIENTO DE LA MALLA AL DEFORMARSE SEGÚN EL ESQUELETO.**

Este problema viene por la rotación no controlada del esqueleto animado, creando en la malla del avatar una deformación. Para solventar este problema, se realizaron diversos cambios. En primer lugar comprobar la asociación de los vértices de la malla del avatar con los huesos del esqueleto, los cuales, afectan al movimiento y modificar esa asociación. En segundo lugar, se realizaron rotaciones extra en huesos que por lo general no son involucrados. Como se indicó en apartados anteriores, cada rotación se realiza por cada eje. Por tanto, los ejes de coordenadas “X” y “Z”, sus rotaciones son de desplazamiento, mientras que la rotación en “Y”, rota a sí mismo. Con esto se consigue, por ejemplo, el movimiento visual de la palma de la mano hacia abajo o hacia arriba; el antebrazo no ha cambiado de posición pero si ha rotado a sí mismo. Con esto, se consigue eliminar esta deformación.

#### 4.4.6. SECUENCIA DE EJECUCIÓN DEL PROGRAMA

Como anteriormente se ha comentado, la aplicación final en realidad son dos aplicaciones trabajando de forma individual: aplicación 3D y aplicación gráfica para el acceso a la base de datos.

La aplicación 3D solo ejecuta una sesión con ejercicios. Los datos serán adquiridos desde la base de datos y se preparará la aplicación para realizar la tarea. Al ser un recorrido secuencial, se opta por hacer una ejecución por pasos, realizando un cambio de paso cuando se cumpla una condición. Con este comportamiento, la programación es más sencilla por dividir toda la ejecución de la aplicación en diferentes pasos, ayudando a la realización de rutinas. Los pasos son los siguientes y (su ejecución se explicará posteriormente):

- **Paso 0:** Este paso inicial realiza la carga de datos de la sesión mientras en pantalla aparece una escena con un mensaje que indica “CARGANDO”. Además se van abriendo y preparando todos los escenarios necesarios.
- **Paso 1:** Finaliza la carga de datos y se ordenan las escenas según la prioridad para mostrar.
- **Paso 2:** Dependiendo del ejercicio, posiciona y prepara el avatar y la cámara.
- **Paso 3:** Preparar el escenario “menú” que es sustituido como pantalla informativa.
- **Paso 4:** Preparar el escenario “HUD” y una animación de una cuenta atrás.
- **Paso 5:** Empieza la animación de cuenta atrás.
- **Paso 6:** Espera a que termine la animación.
- **Paso 7:** Activar el escenario “HUD” para mostrar los textos que aparezcan en pantalla.
- **Paso 8:** Ejecución del ejercicio, contar tiempo, actualizar datos del “HUD” y actualizar ángulos.
- **Paso 9:** Comprobar si termina una serie, un ejercicio o termina la sesión. Realizar los ajustes para continuar.
- **Paso 10:** En caso de continuar con el mismo u otro ejercicio: realizar animación para volver al escenario informativo y desaparecer el escenario “HUD”.
- **Paso 11:** En caso de terminar la sesión: mostrar el escenario de carga con un texto que indique “GUARDANDO”.
- **Paso 12:** Calcula los tiempos, guarda la información en la base de datos.
- **Paso 13:** Salir de la aplicación 3D.

Estos son los pasos planteados para la ejecución de la aplicación 3D. Ahora se va a explicar paso a paso los del contenido de cada proceso.



IMAGEN 28: PANTALLA DEL ESCENARIO CARGA

El escenario de carga es el primero que se ejecuta nada más se ejecuta la aplicación. El script que contiene la máquina de estados debe ejecutarse en este escenario. Al inicio, sólo hay un escenario activo, el que sale por la pantalla con el texto “CARGANDO” como se muestra en la imagen 28. El paso 0 consiste en comunicarse con la base de datos. Para esto, Blender necesita acceder a la librería de “MySQL”, el cual, no dispone de forma nativa. Para ello se requiere la instalación del Connector/Python de MySQL, pero este conector no se puede instalar sin un IDLE de Python instalado (Entorno de desarrollo integrado). El motivo viene porque Blender,

aunque utilice Python como lenguaje, Blender utiliza una versión limitada y exclusiva para esta herramienta. Con la ayuda de la comunidad, se encuentra una solución para el problema.<sup>14</sup> En el equipo utilizado para el desarrollo de la aplicación se instala el IDLE con versión 3.4.3 de “Python”. Se realiza la instalación del conector de MySQL. Sin este conector, “Python” no dispondrá de la librería que dé acceso a la base de datos. Para que el “Python” de Blender pueda disponer de la librería de MySQL, se accede a la carpeta de la herramienta “Python” y se copia las siguientes carpetas: DLLs, y site-packages. Estas carpetas se guardan en una ruta accesible para la aplicación, y que no cambie. En Blender, en el código del script, se llama a la función `sys.path.append()` de “Python” para asociar estas carpetas con las propias de Blender. Con esto, ya se pueden importar las librerías de MySQL en el Python de Blender.

Para realizar una única consulta a la base de datos, se crea una propiedad Booleana en la cámara de la escena. Esta propiedad, mediante la condición “if”, activa las funciones que realizan la consulta. Una vez terminada la consulta, cambia el valor de la propiedad bloqueando que se pueda hacer una nueva consulta, ya que los scripts que utiliza Blender son ejecutados en cada imagen procesada, y evitar que la ejecución de la aplicación sea lenta. Además que realizando una consulta a la base de datos es suficiente.

En el **paso 0**, adquirir el número de sesión y el DNI del paciente en la tabla “importar”. Como sólo hay una fila con éstos datos, no hay posibilidad de equivocación. Con esta información ya se puede acceder a los datos de la sesión. Con el DNI y el número de sesión, se encuentran los ejercicios y el sexo del paciente para que el avatar sea acorde al sexo almacenado. Una vez obtenida toda la información, se guardan en propiedades de un objeto del escenario. En este caso, la cámara contiene las propiedades variables tales como: la sesión completa, los ejercicios completos, contador de ejercicios, contador de series, temporizadores, número de ejercicios máximo, y otros secundarios. El objeto utilizado como fondo de pantalla almacena con sus propiedades los ejercicios desglosados por datos. En este paso también se ejecuta la carga y activación de los demás escenarios. Al ser un proceso que requiere de más tiempo de procesamiento puede crea algunos frames que ocupan más tiempo del necesario para un frame (si se prepara para 60 frames, quiere decir que como máximo debe de durar 16 ms un frame), se prepara un temporizador para dar tiempo a la carga. Al ejecutarse en cada imagen todo el código completo, no se recomienda utilizar bucles como: “for”, “while”, etc... porque estos se ejecutarían en cada frame y ralentizarían la aplicación. Para eso, se utiliza una propiedad de tipo entero la cual irá sumando una unidad cada vez que se ejecute el código (cada frame), hasta llegar a una condición controlada por un “if”. Cuando ésta condición sea positiva, la carga de los escenarios se habrá completado y se puede avanzar al paso siguiente.

En el **paso 1** es la configuración del entorno 3D, el avatar a utilizar y el ejercicio a ejecutar. En este paso se analiza el sexo configurado del paciente y se prepara el avatar masculino o el femenino. Se configura el script de control del avatar y se posiciona el avatar a utilizar en el centro del cubo. Se crea una matriz donde guardar los tiempos por ejercicios y series, y otra matriz para almacenar el ángulo máximo de cada ejercicio. En el último paso, estos datos serán enviados a la base de datos.

En el **paso 2** se posiciona la cámara, y se prepara el avatar para la realización del ejercicio. Este paso es sencillo y con una vez que se ejecute es suficiente.

---

<sup>14</sup> Web de la comunidad: [stackoverflow.com](https://stackoverflow.com)

El **paso 3** realiza la preparación de la figura y textos que aparecen en el escenario “Info previo”. Este escenario muestra la información que aparecerá antes de empezar el ejercicio. Esta información será la mostrada en la siguiente figura.



IMAGEN 29: PANTALLA INFORMATIVA ANTES DE EMPEZAR EL EJERCICIO.

Como se muestra en la imagen 29, hay un indicador del número de ejercicio, la serie, una figura que indica el movimiento y una frase escrita por el fisioterapeuta. El número de ejercicio irá variando cada vez que se complete un ejercicio. En la serie indica el número de serie que se va a ejecutar y la totalidad de series que debe repetir el paciente. La figura que se muestra en la pantalla informativa, en realidad hay un total de 10 figuras, 5 para cada brazo. En el caso del ejercicio flexión glenohumeral dispone de tres figuras demostrativas. En este ejercicio hay la posibilidad de levantar el brazo únicamente hacia delante, también puede realizar el movimiento únicamente hacia atrás o incluso combinar los dos movimientos. Por ello, para cada opción se ha creado una figura distinta. Por último, también aparece un texto escrito por el fisioterapeuta, que puede ser cualquier escrito que quiera recordar o indicar al paciente antes de iniciar el ejercicio. El motor de renderizado “Blender Game” no dispone de una función que centre el texto. Se ha creado una pequeña función propia que, dependiendo de los caracteres incluidos, desplaza hacia la izquierda el texto creando un efecto de centrado.

```
def Centrar(texto, objeto, y):  
    objeto.text = texto  
    textScaleX = len(texto)  
    objeto.localPosition[1] = y - objeto.localScale[1]/2  
    objeto.localPosition[0] = -len(texto) * (0.3 * objeto.localScale[0])
```

IMAGEN 30: CODIGO UTILIZADO PARA REALIZAR EL CENTRADO DEL TEXTO.

Una vez configurada toda la información a mostrar en este escenario es momento de cambiar de paso. En un futuro, la figura mostrada, puede convertirse en una animación de cada ejercicio realizando el movimiento exactamente.

Para el **paso 4**, se prepara la información que aparece cuando el ejercicio se está ejecutando. Esta información se establece en el escenario “Escena HUD”. Consiste en cambiar la resolución del texto a mostrar. Porque si no se hace, el texto sale pixelado.

En el **paso 5**, desaparece la pantalla de carga para que aparezca la información previa al ejercicio y de fondo el avatar situado y preparado. Para no desactivar el escenario de carga que es el que está ejecutando la máquina de estados, la cámara de este escenario cambia de posición para no observar ni el texto ni el fondo negro utilizado. De esta forma, no hay ningún objeto superpuesto en este escenario, aun estando activado.

En el **paso 6**, ya se activa la animación “cuenta atrás”. Esta animación se emplea para empezar la ejecución del ejercicio después de un cierto tiempo. Contiene una cuenta atrás, de 5 a 0 segundos. Antes que esta cuenta atrás, está 10 segundos de espera, para que el paciente pueda leer la información. Cuando esta animación termine, se pasa al séptimo paso.

En el **paso 7**, cuando ya se ha terminado la animación de la cuenta atrás, se activa la animación de retirada de la pantalla con la información a un lado. La cámara del escenario “info” cambia de ubicación para distanciarse de los objetos y dar la sensación de que son los objetos los que se hacen pequeños y se desplazan a un lateral. Este cambio se hace con una animación para que el movimiento sea fluido. La pantalla con la figura se quedan a la esquina superior izquierda y los textos desaparecen. Una vez terminada la animación, pasa al siguiente paso.

En el **paso 8**, es el momento que el paciente realizará el ejercicio. Nada más se ejecute por primera vez, activa la visualización de los textos del escenario “HUD”. Activa también el script del avatar bloqueado por un “if”, para poder ejecutar su funcionamiento. Un sensor “Timer” es reiniciado para utilizarlo como cronómetro, se limita la visualización a minutos y segundos realizando las siguientes operaciones.

```
Tiempo60s = Tiempo-int(Tiempo/60)*60
if Tiempo60s < 10: Segundos = '0' + str(Tiempo60s)
else: Segundos = str(Tiempo60s)
MinutoSegundo = str(int(Tiempo/60)) + ':' + Segundos
```

IMAGEN 31: CODIGO PARA CONVERTIR EL TIEMPO DEL CRONOMETRO EN UN STRING DE MINUTOS Y SEGUNDOS.

Los datos variables que se almacenan en el objeto del escenario “Cargando” se envían a los objetos del escenario “HUD” para que muestren la información. Los datos concretos son el número de repeticiones, el ángulo de trabajo y el ángulo máximo conseguido. Además de modificar el valor del ángulo máximo cada vez que el paciente lo supere. El objeto esqueleto del escenario “3D” se encarga de realizar los ejercicios, calcular los ángulos y ejecutar el movimiento del avatar. Cuando las repeticiones lleguen a igualar o superar el máximo programado, salta la condición para cambiar al siguiente paso.

En el **paso 9** se considera que ya ha terminado el ejercicio. Posiciona el avatar en su punto neutro y deshabilita el código utilizado para la ejecución del ejercicio de forma que el avatar no realice ningún otro movimiento. Compara el número de la serie, si este es inferior al máximo indicado para el ejercicio, añade una unidad a la serie y pasa al siguiente paso; si es el máximo de la serie, significa que cambia de ejercicio, resetea el número de la serie, realiza la función para buscar el siguiente ejercicio y cambia al siguiente paso; y, por último, si no es ninguno de los anteriores y, además, ha llegado al final de todos los ejercicios, salta al paso 11.

En el **paso 10**, se encuentra la activación de una animación de retroceso de la pantalla informativa. Esta animación es la inversa al movimiento del paso 7. Se ocultan los textos del escenario “HUD”, la pantalla en pequeño se expande hasta ocuparlo todo y activa los textos. Cuando termine la ejecución de la animación cambia al paso 2 y comienza de nuevo con un ejercicio o serie diferente.

En el **paso 11** se ejecuta cuando se han terminado todas las series y ejercicios programados. La única tarea es visualizar un fondo negro y un texto indicando que está guardando mediante el siguiente código.

```
PropCamEsceCarga.localOrientation = [-math.pi/2,0,0]
PropCamEsceCarga['Prop Escenario INFO'].objects['Camara Escena INFO'].worldOrientation = [math.pi/2,0,0]
PropCamEsceCarga['Prop Escenario HUD'].objects['Camara Esce HUD'].worldOrientation = [0,0,math.pi]
```

IMAGEN 32: CODIGO PARA LA COLOCACIÓN DE LAS DIFERENTES CAMARAS PARA MOSTRAR EN LA PANTALLA EL MENSAJE “GUARDANDO”.

Éste paso y los dos siguientes podrían ir juntos, pero se decidió separar las tareas visuales hacia el paciente y las tareas internas como la memorización en la base de datos.

En el **paso 12**, se realiza todo el cálculo y almacenamiento de la información en la base de datos. En primer lugar, en la matriz que almacena los tiempos, estos se encuentran en segundos. Por tanto, se cambia al formato reconocido por SQL (Hora, Minuto, Segundo). Como puede haber varios ejercicios, se crea un bucle para ir realizando las modificaciones de los tiempos en cada ejercicio ejecutado. Con esto, la aplicación 3D podría ralentizarse por el tiempo que necesite para realizar las modificaciones en la base de datos, pero como en el paso anterior ya se cambió la pantalla y el texto por el de “GUARDANDO”, al usuario no notará que la aplicación se ha quedado congelada. Al igual que se almacena la información de los tiempos, también se almacena el ángulo máximo alcanzado por el paciente. Terminada la modificación de los ejercicios en la base de datos, se realiza un cálculo total del tiempo transcurrido en la sesión y se modifica el tiempo de sesión y el booleano que indica la ejecución de esta sesión. Si no hay ningún error, cambia al siguiente paso.

En el **paso 13**, da un pequeño tiempo de 5 frames (equivalente a 8.33 ms) antes de salir de la aplicación. Éste pequeño tiempo es de precaución por posibles problemas con los tiempos de ejecución de la aplicación 3D y de la base de datos.

#### 4.4.7. SCRIPTS DE CADA ESCENARIO

Otros scripts que se ejecutan en algún momento, son los propios de cada escenario. Ya sea para configurar, o controlar una animación, estos son necesarios para el buen funcionamiento de la aplicación.

##### 4.4.7.1. SCRIPT UTILIZADO EN EL ESCENARIO “INFO”

Este script configura la ubicación del texto que aparece antes de iniciar el ejercicio y, además, cambia el tamaño de la figura mostrada dependiendo de si se está ejecutando o no el ejercicio. Esto se debe a que, durante la muestra de la información la figura tiene un tamaño menor para dejar espacio al texto. Cuando la pantalla se queda a un lateral en pequeño, el texto desaparece reubicándose en un lugar fuera del objetivo de la cámara. Con ello, la figura puede ser más grande y verse mejor. Este script contiene una condición para que se ejecute o no, dependiendo de una propiedad booleana. Esta propiedad es modificada en el paso 7 (activar) y 10 (desactivar).

##### 4.4.7.2. SCRIPT QUE CONTROLA EL ESCENARIO “HUD”

Este script posiciona el texto a mostrar. Al igual que en el script anterior, una propiedad controla internamente su ejecución y que debe hacer en cada momento. Con ello, en el momento que la máquina de estados indique que debe mostrar los textos en pantalla, éste script ubicará cada texto en el lugar correspondiente. Y cuando no, los cambiará de ubicación para no ser mostrados.

##### 4.4.7.3. SCRIPT QUE CONTROLA AL AVATAR Y LOS EJERCICIOS

Este script se ejecuta en el escenario “3D”. Recibe la información de las articulaciones controladas por el dispositivo Kinect V2 y la procesa para mostrar el movimiento en el avatar. Una propiedad booleana y otra entera, controlan la ejecución de las líneas código mediante la condición “if” dependiendo del ejercicio a realizar. Como ya se explicó anteriormente, cada sub-apartado realiza un cálculo vectorial de las articulaciones. Con este cálculo, se obtiene el ángulo real que se mostrará por pantalla y que se aplica al avatar. Este ángulo calculado se convierte en un valor para ser utilizado en la animación preparada y es almacenado en una propiedad propia del esqueleto del avatar. Mediante el actuador de animación, establece el frame con el ángulo calculado. El avatar posiciona su movimiento igual que el usuario delante del dispositivo Kinect V2.

## 4.5. BASE DE DATOS MYSQL

Para crear una nueva base de datos, hay que tener claro el alcance de ésta. La aplicación final de este proyecto tiene como objetivo estar instalado en un equipo individual y específico para dicho trabajo. La instalación de la base de datos es exclusivamente del ordenador donde esté instalada la aplicación. Se debe configurar la conexión que se utilizará para acceder a la base de datos (en este caso es una conexión local), el nombre y una contraseña. Con esto, la base de datos está creada a falta de introducir todas las tablas y uniones. Como ya se indicó, “SQL” y sus variantes, crea un servidor que se comunica con otras aplicaciones, permitiendo acceder y modificar el contenido de la base de datos.

Una vez creado el usuario para trabajar con “MySQL”, es el momento de crear las tablas y las conexiones entre ellas. Se necesitan las siguientes tablas:

- La tabla de “pacientes” almacena toda la información necesaria del paciente: nombre, apellidos, edad, sexo, teléfono, DNI, lesión, fecha de la lesión, inicio y finalización del tratamiento, número de registro, una contraseña y otros datos de interés.
- La tabla de “sesiones” contiene los datos de la sesión antes y después de ser realizada: conector a un paciente<sup>15</sup>, nombre de la sesión, número del orden a ejecutar, fecha y duración, y un campo que indique si se ha ejecutado.
- La tabla de “ejercicios” almacena toda la configuración de cada ejercicio. De esta forma, cada ejercicio puede ser único y exclusivo: conector a una sesión<sup>10</sup>, conector a un ejercicio virtual<sup>10</sup>, nombre del ejercicio, número de repeticiones, número de series, número del orden a ejecutarse, ángulo a conseguir, ángulo máximo alcanzado por el paciente, ángulo mínimo donde empezar las repeticiones, un texto que aparezca en pantalla, si es el brazo izquierdo o derecho y la duración del ejercicio.
- La tabla de “ejercicios virtuales” contiene los ejercicios que se pueden realizar con la aplicación 3D. Actualmente, dispone de tres ejercicios para la rehabilitación del hombro. Esta tabla contiene: nombre del ejercicio y número del ejercicio.
- La tabla de “fisioterapeutas” fue añadida posteriormente al diseño inicial, aun así, es importante para un mejor manejo de la aplicación. Contiene lo siguiente: Nombre y apellidos del profesional, DNI, un indicador si es administrador en la aplicación (y por tanto poder realizar otras gestiones), y una contraseña.
- Por último, la tabla “importar” tiene un uso interno. Como su nombre indica, su uso es para importar la información desde la base de datos hasta la aplicación 3D y conocer qué sesión debe ejecutar. Como las tablas contienen mucha información, se necesita de un lugar de toda la base de datos que sea fija, y de esta forma, la aplicación 3D tiene un lugar donde realizar una consulta. Cada vez que se vaya a ejecutar dicha aplicación, en la primera fila de esta tabla se modificará la información respecto a la sesión a ejecutar, de este modo, siempre habrá una única fila. Esta tabla solo dispone de dos campos: el conector de la sesión<sup>10</sup> y el DNI del paciente. Más adelante se explicará el porqué de almacenar el DNI del paciente.

Hay diferentes formas de realizar conexiones entre tablas. Para este caso solo se utiliza una de ellas, la denominada “1:n”. Cada fila de cada tabla es un dato; sabiendo esto, “1:n” indica que “1”, una fila de la primera tabla, puede ser conectado a “n”, muchas filas de la segunda tabla. En este caso, un paciente puede tener muchas

---

<sup>15</sup> Los conectores de tablas son un número de índice que indica una fila exclusiva de la tabla a conectar.

sesiones, al igual que una sesión puede tener muchos ejercicios y un ejercicio virtual puede estar configurado en muchos ejercicios.

Con esto, la relación de las tablas a utilizar es la que se muestra en la siguiente figura:

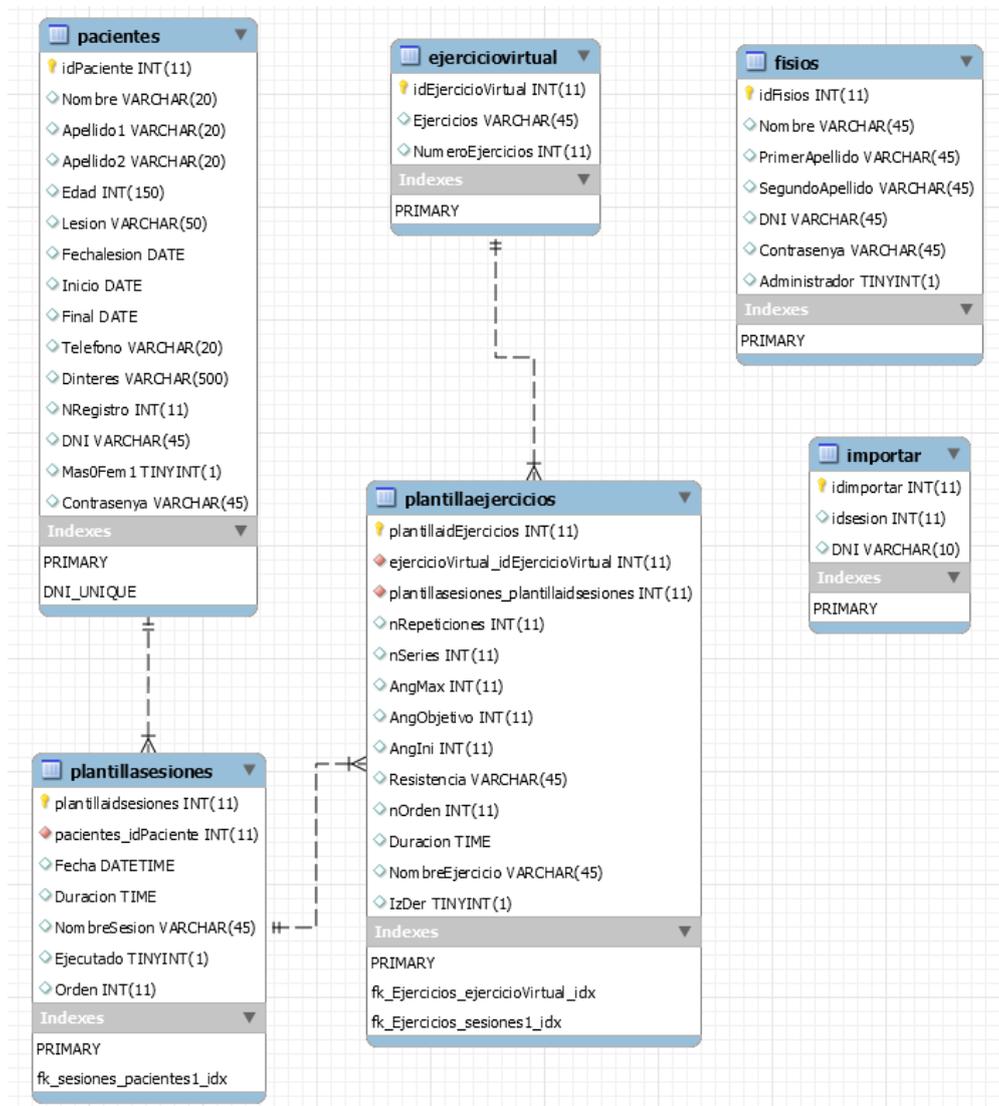


IMAGEN 33: RELACIÓN DE LAS TABLAS DE LA BASE DE DATOS.

Posteriormente a la creación de esta base de datos, se hizo un cambio de la lógica de las tablas “sesiones” y “ejercicios”. Este cambio consistió en crear tablas de las anteriores nombrada por cada paciente almacenado. Con esa variación, cada paciente dispone su propia tabla de “sesiones” y su tabla de “ejercicios”.

A la hora de hacer una consulta a una base de datos, ésta trabaja comparando la información enviada a través de la consulta, con los datos de la tabla, fila a fila. Al inicio no habría problemas, ya que las tablas contendrían pocas filas de información. Por ejemplo: si en un día hubiese 3 sesiones, y cada una contienen 5 ejercicios, en un mes, podría haber más de 300 ejercicios; en un año alrededor de los 4.000. Hacer una consulta donde hay 4.000 filas, y además tiene que comparar de fila en fila, tardaría mucho tiempo en procesar la consulta. De este modo, cada paciente tendrá su tabla y aun estando un año con tratamientos (no es lo normal), no llegaría a superar los 1.000 ejercicios.

Para una creación de tablas más sencilla, el nombre y conexiones de las tablas contendrán el DNI del paciente. Todas las tablas “sesiones” se enlazan a la tabla

“paciente”, cada tabla “ejercicios” enlaza con la tabla “sesiones” con el mismo DNI; y todas las tablas “ejercicios” enlazan con la tabla “ejercicios virtuales”.

Cada vez que se crea un nuevo paciente en la tabla “pacientes”, se crean las dos nuevas tablas (“sesiones” y “ejercicios”). Si se borrase el paciente, las tablas también se borrarán. Con este cambio, el rendimiento de las consultas ha mejorado.

## **4.6. APLICACIÓN GRÁFICA DE USUARIO Y DE ACCESO A LA BASE DE DATOS**

Para el proyecto se ha desarrollado una aplicación en lenguaje C# que accede a la base de datos MySQL. Nada más se crea el proyecto en Visual Studio, se importa la librería de MySQL. Como pasa en “Python”, se necesita instalar un conector para que Visual Studio tenga acceso a la librería de MySQL. El conector a instalar es MySQL/NET. Con esto, cualquier herramienta o aplicación de Microsoft, podrá acceder a la base de datos. Dentro del proyecto desarrollado en Visual Studio, se accede a las referencias y se agrega la librería MySQL. Con esto, ya se puede utilizar las librerías de MySQL.

Se han creado las siguientes clases en el proyecto, para el desarrollo completo de la aplicación:

- Clases visuales (Forms):
  - o Formulario principal.
  - o Formulario crear/modificar paciente.
  - o Formulario paciente.
  - o Formulario ver/crear/modificar sesión.
  - o Formulario ver/crear/modificar ejercicio.
  - o Formulario empezar sesión.
  - o Formulario buscar.
  - o Formulario crear/modificar trabajador (Añadido posteriormente).
  - o Formulario Login (Añadido posteriormente).
  - o Formulario contraseña (Añadido posteriormente).
- Clases internas:
  - o Conector MySQL.
  - o Pacientes
  - o Sesiones
  - o Ejercicios
  - o Ejercicios virtuales
  - o Controlador de formularios
  - o Otras operaciones.
  - o Adjuntos. Posteriormente se modificó por Conexión externa.
  - o Trabajadores.

### **4.6.1. EL FORMULARIO “LOGIN”**

A la hora de ejecutar la aplicación, antes que nada se tiene que conocer qué usuario va a acceder, tanto paciente como fisioterapeuta. Para ello inicialmente aparecerá una ventana con el formulario “Login” (login form). Este formulario dispone de dos “CheckBox” para seleccionar si se trata de un empleado o un paciente; dos cuadros de texto para introducir el DNI y la contraseña. Si todo es correcto, se abrirá la aplicación.

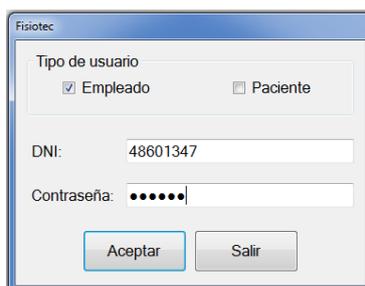


IMAGEN 34: VENTANA DE INICIO DE LA APLICACIÓN.

#### 4.6.2. FORMULARIO “PRINCIPAL”

Éste formulario es el encargado de ser el marco principal de toda la aplicación. En él, se alojarán los demás formularios de forma que parezca todo una ventana fija. Los formularios “usuario”, “buscar” y “contraseña”, se abren en diferente ventana. En el formulario “principal” se inserta una barra de menú con distintas opciones disponibles. Estas opciones son las de abrir los demás formularios, y salir de la aplicación. Para que haya un orden lógico, algunas opciones del menú serán inhabilitadas dependiendo de que se esté haciendo. De esta forma, no se podrán crear, modificar o borrar sesiones o ejercicios sin seleccionar al paciente previamente.

Otro motivo para inhabilitar las opciones del menú es para diferenciar entre usuarios. Un paciente no puede acceder a crear sus propios ejercicios.



IMAGEN 35: FORMULARIO "PRINCIPAL".

La imagen 36 muestra los diferentes submenús incluidos en cada desplegable. Indicar que los menús “Sesiones” y “Ejercicios” son iguales que el menú “Pacientes”. Se indicará en cada formulario como se puede acceder a ellos.

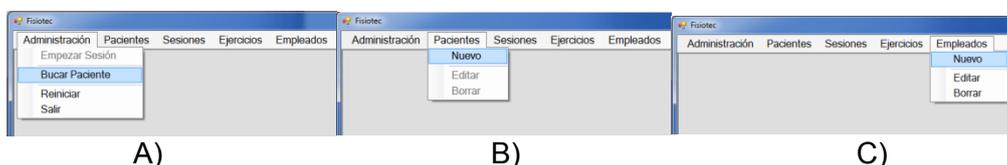


IMAGEN 36: A) MENU ADMINISTRACIÓN DESPLEGADO, B) MENÚ PACIENTE DESPLEGADO, C) MENÚ EMPLEADO DESPLEGADO.

#### 4.6.3. FORMULARIO “CREAR/MODIFICAR PACIENTE”

Este formulario dispone de dos modos de visualizar.

1. En el modo “crear”, se puede acceder desde el menú desplegable “Pacientes” > “Nuevo”, y a través del formulario “Paciente” (sección “Paciente”, botón “Nuevo”).

- Para acceder al modo “modificar”, se puede acceder a través del formulario “Paciente” (sección “Paciente”, botón “Editar”) y desde el menú desplegable “Paciente” > “Editar” cuando se esté visualizando el formulario “Paciente”.

Contiene cuadros de texto para rellenar los datos personales e información del paciente. Además, dispone de dos botones: uno realiza la función de crear/modificar los datos (enviando la información a la base de datos), y el otro para cancelar la creación o modificación del paciente. Este formulario, para ser creado, recibe información sobre si se ejecuta para crear un paciente nuevo o por el contrario es para modificar uno ya existente. En este último caso recibe la información del paciente y la introduce en cada cuadro de texto.

IMAGEN 37: FORMULARIO CREAR/MODIFICAR UN PACIENTE

#### 4.6.4. FORMULARIO “PACIENTE”

Este formulario se podrá acceder a través del buscador y seleccionando a un paciente.

Contiene la información a mostrar del paciente, una lista de archivos adjuntos para el paciente y otra de las sesiones programadas. Todos los cuadros de texto están en modo “solo lectura”. Las listas disponen de botones para su interacción. En el caso de la lista de los archivos adjuntos, contiene tres botones: el botón *Ver archivo*, una vez seleccionado una fila de la lista, abre el fichero adjunto; el botón *Adjuntar*, abre una ventana dialogo que le permite navegar entre los ficheros hasta seleccionar un archivo, este archivo se copia a una carpeta propia del programa para tener acceso posteriormente; y el botón *Borrar* que, con una fila de la lista seleccionada, elimina el archivo de la propia carpeta y de la lista.

Ejecutado	NombreSesion	Orden	Fecha
<input checked="" type="checkbox"/>	Calentamiento	1	00:00:00 01/01/
<input type="checkbox"/>	Poten. flexión	2	00:00:00 01/01/
<input type="checkbox"/>	De todo	3	00:00:00 01/01/

IMAGEN 38: FORMULARIO PARA VISUALIZAR LOS DATOS DE UN PACIENTE.

La lista de sesiones dispone de las siguientes opciones en forma de botones: el botón *Ver sesión*, abre otro formulario con la información de la sesión seleccionada de la lista; el botón *Nueva sesión*, abre otro formulario para crear de cero una sesión; el botón *Editar*, de la sesión seleccionada, abre un formulario para poder editar la información; el botón *Borrar*, borra la sesión y si contiene ejercicios, los borra también de la base de datos; el botón *Importar*, abre el buscador para buscar las sesiones almacenadas en la plantilla una vez seleccionado, se copia tal cual en la lista del usuario; el botón *Exportar*, de la sesión seleccionada de la lista, realiza una copia a la plantilla. Además, en esta sección dispone de dos botones para reordenar la lista de sesiones según el interés del fisioterapeuta a la hora de ejecutar en primer lugar una u otra sesión.

En este formulario también dispone de una sección con las siguientes opciones: el botón *Crear nuevo paciente*, abrirá un nuevo formulario para crear un paciente nuevo; el botón *Editar*, abre un nuevo formulario con los cuadros de texto habilitados y para modificar los datos del paciente; el botón *Borrar*, borra toda la información del paciente abierto seleccionado, además, si contiene sesiones y ejercicios, también los elimina; el botón *Establecer contraseña*, abre una ventana donde poder establecer o cambiar la contraseña de entrada. Dependiendo de quién abra este formulario, el paciente o el fisioterapeuta, se habilitarán o inhabilitarán ciertos botones. Estos botones son los siguientes: para el fisioterapeuta, todos los botones estarán visibles y habilitados; para el caso del paciente, solo dispondrá de los botones de *Establecer contraseña*, *Ver archivos adjuntos* y *Ver sesión*.

#### 4.6.5. FORMULARIO “VER/CREAR/MODIFICAR SESIÓN”

Este formulario dispone de tres modos de interacción. Para poder acceder, requiere como mínimo tener abierto el formulario “Paciente”.

1. Para entrar en modo “ver” se puede acceder a través del formulario “Paciente” (sección “Lista de sesiones”, botón “Ver”).
2. En el modo “crear”, se puede acceder desde el menú desplegable “Sesiones” > “Nuevo”, y a través del formulario “Paciente” (sección “Lista de sesiones”, botón “Nueva”).
3. Para acceder al modo “modificar”, se puede acceder a través del formulario “Paciente” (sección “Lista de sesiones”, botón “Editar”) y desde el menú desplegable “Sesiones” > “Editar” cuando se esté visualizando el formulario “Sesión” en modo “Ver”.

Es el encargado de mostrar, crear y editar una sesión. Dispone de una pequeña sección con información del paciente, otra sección con los datos de la sesión, una tercera sección con una lista de ejercicios y un botón para abrir el formulario “empezar sesión”.

Datos del paciente

Nombre:  N° Registro:

Primer apellido:

Segundo apellido:  Lesión:

Datos de la sesión

Nombre de la sesión:  Sesión N°:

Fecha de realización:  Duración:

Lista de ejercicios

Ver Ejercicio	NombreEjercicio	NOrden	Duracion	ID
<input checked="" type="checkbox"/>	Levantamiento lateral	1	00:00:01	2
<input type="checkbox"/>	levantar brazo derecho	2	00:00:00	3
<input type="checkbox"/>	Brazo lado derecho	3	00:00:00	4
<input type="checkbox"/>	Izquierda 45º	4	00:00:00	5
<input type="checkbox"/>	Rotación Derecha 10º	5	00:00:00	6
<input type="checkbox"/>	Derecha 90º	6	00:00:00	7

Empezar Sesión

Atras

IMAGEN 39: FORMULARIO PARA VISUALIZAR LA INFORMACIÓN DE UNA SESIÓN.

En primer lugar, la sección con los datos del paciente, son cuadros de texto en modo “solo lectura” que informan y recuerdan a que paciente se refiere esta sesión.

De los datos de la sesión, únicamente puede ser editado el nombre, ya que, los demás campos son rellenados automáticamente, y en el momento que oportuno. Cuando se crea una sesión, el campo del número de sesión es escrito automáticamente, mientras que los campos de la fecha de realización, la duración y el texto “EJECUTADO” se rellenarán una vez terminado de ejecutar la sesión con la aplicación 3D.

La lista de ejercicios dispone de diversas opciones dependiendo de quién ejecute el formulario, al igual que ocurre en la lista de sesiones del formulario “pacientes”. Dispone de los siguientes botones: el botón *Ver ejercicio*, abre un nuevo formulario con la información del ejercicio; el botón *Nuevo ejercicio*, abre un nuevo formulario para programar un nuevo ejercicio; el botón *Editar*, abre un formulario con la información del ejercicio seleccionado para poder ser editado; el botón *Borrar*, borra el ejercicio seleccionado; *Importar*, abre el buscador dentro de la plantilla de ejercicios y copiar el ejercicio seleccionado a la sesión; el botón *Exportar*, del ejercicio seleccionado de la lista, realiza una copia a la plantilla. Además, dispone de los botones para reordenar los ejercicios según desee el fisioterapeuta. Cuando sea ejecutado en el entorno 3D, se ejecutará con dicho orden.

#### 4.6.6. FORMULARIO “VER/CREAR/MODIFICAR EJERCICIO”

Este formulario dispone de tres modos de interacción. Para poder acceder, requiere como mínimo tener abierto el formulario “Sesión”.

1. Para entrar en modo “ver” se puede acceder a través del formulario “Sesión” (sección “Lista de ejercicios”, botón “Ver”).
2. En el modo “crear”, se puede acceder desde el menú desplegable “Ejercicios” > “Nuevo”, y a través del formulario “Sesión” (sección “Lista de ejercicios”, botón “Nuevo”).
3. Para acceder al modo “modificar”, se puede acceder a través del formulario “Sesión” (sección “Lista de ejercicios”, botón “Editar”) y desde el menú desplegable “Ejercicios” > “Editar” cuando se esté visualizando el formulario “Ejercicios” en modo “Ver”.

Dentro de este formulario dispone de las siguientes secciones: Datos del paciente, donde recuerda información parcial del paciente afectado; datos de sesión, que

contiene información parcial de la sesión; datos del ejercicio, que contiene todos los apartados necesarios para el ejercicio.

Los datos del paciente y sesión son cuadros de texto en modo “solo lectura” que recuerdan el paciente y la sesión de este ejercicio.

Datos del paciente	
Nombre:	Rafa
Nº Registro:	1
Primer apellido:	Conejero
Segundo apellido:	Vila
Lesión:	Ligamentos cruzados

Datos de la sesión	
Nombre de la sesión:	Calentamiento
Sesión Nº:	1

Datos del ejercicio	
Nombre del ejercicio:	Brazo lado derecho
Ejercicio base:	Brazo hacia al lado
Series:	1
Orden:	3
Repeticiones:	2
Duración:	
Ángulo inicial:	0
Descripción:	Peso de 1 Kg.
Ángulo objetivo:	90
Ángulo máximo:	180

Izquierdo <-- Miembro -->  Derecho

Atras

IMAGEN 40: FORMULARIO PARA VISUALIZAR UN EJERCICIO.

Este formulario contiene cuadros de texto, dos “CheckBox”<sup>16</sup> y un desplegable. Los cuadros de texto son para el nombre del ejercicio, el número de series y repeticiones de cada serie, el ángulo inicial, objetivo y máximo (este último en modo “solo lectura”), que determina el recorrido del ejercicio y una vez terminada la ejecución de la sesión almacenar el ángulo máximo alcanzado por el paciente. Además, otro cuadro de texto para almacenar una descripción que se mostrará al paciente antes de empezar a ejecutar el ejercicio. Los cuadros duración y orden están también en modo “solo lectura” ya que su información es automatizada. En el caso del número de orden, éste es escrito cuando se crea por primera vez un ejercicio; mientras que la duración se completará cuando termine de ejecutar la sesión. Por último, dispone de dos “CheckBox” para seleccionar el brazo izquierdo o el derecho.

#### 4.6.7. FORMULARIO “EMPEZAR SESIÓN”

Para poder acceder a este formulario, requiere tener abierto el formulario “Sesión”. Se puede acceder desde el menú “Administración” > “Empezar Sesión” y desde el botón verde “Empezar sesión” situado a la izquierda en el formulario “Sesión”.

Aparece desde el formulario de la sesión que se va a ejecutar. En éste formulario, dispone de diferentes cuadros de texto con la información del paciente y de la sesión a ejecutar. Un botón con el texto “empezar” ejecuta el entorno 3D de la sesión indicada.

<sup>16</sup> CheckBox es una casilla de selección, empleado en programación visual

Datos del paciente

Nombre:  Nº Registro:

Primer apellido:

Segundo apellido:

Datos de la sesión

Nombre de la sesión:  Sesión Nº:

**EMPEZAR**

Atras

IMAGEN 41: FORMULARIO PARA EMPEZAR A EJECUTAR UNA SESIÓN.

#### 4.6.8. FORMULARIO “BUSCADOR”

Este formulario dispone de tres modos de interacción. Para buscar pacientes, buscar sesiones de la plantilla, buscar ejercicios de la plantilla y buscar trabajadores.

1. Para buscar pacientes acceder desde el menú “Administración” > “Buscador de pacientes”.
2. Para buscar sesiones de la plantilla, desde el formulario “Paciente”, en la sección “Lista de sesiones”, el botón “Importar”.
3. Para buscar ejercicios de la plantilla desde el formulario “Sesión”, en la sección “Lista de ejercicios”, el botón “Importar”.
4. Para buscar trabajadores, en el menú “Empleados” > “Editar” y “Empleados” > “Borrar”.

Para optimizar el trabajo del fisioterapeuta, se creó un formulario que se abrirá en una ventana aparte, llamada “**buscador**”. En él, el fisioterapeuta podrá acceder a todos los pacientes que tenga almacenado en la base de datos.

Contiene dos secciones, en primer lugar opciones de búsqueda donde un desplegable indica que buscar. Un ejemplo sería buscar por nombres o por DNI. Se añade opciones lógicas a la búsqueda que activarán otras partes de las opciones de búsqueda. Como se muestra en la siguiente imagen, un desplegable permite seleccionar el tipo de lógica a emplear en la búsqueda. Dispone de las operaciones lógicas Y (AND) y O (OR).

Buscador de Pacientes

Nombre  Primer apellido

Buscar

Resultados de la búsqueda

	NRegistro	Nombre	Apellido1	Apellido2	Edad	DNI	Le
▶	1	Rafa	Conejero	Vila	29	48601347	Lic

Aceptar Cancelar

IMAGEN 42: VENTANA DEL BUSCADOR DE EJERCICIOS

Además, a la hora de realizar una búsqueda, no requiere de una palabra o texto completo, se puede realizar búsquedas parciales. De esta forma, facilita la búsqueda del paciente.

Este mismo formulario se utiliza para realizar las importaciones de sesiones y ejercicios. Utiliza otras líneas de código para que solo acceda a las plantillas configuradas por el propio fisioterapeuta.

#### 4.6.9. FORMULARIO “CREAR/MODIFICAR TRABAJADOR”

Este formulario dispone de dos modos de interacción. Para crear un trabajador nuevo y para editar un trabajador.

1. Para entrar en modo “crear” acceder desde el menú “Empleados” > “Nuevo”.
2. Para entrar en modo “modificar” acceder desde el menú “Empleados” > “Editar”, realizar la búsqueda desde el formulario “Buscador”.

Es un formulario utilizado en el caso de que el centro de tratamientos disponga de diferentes empleados. De esta forma, se pueden añadir empleados que puedan acceder al programa. Los empleados pueden ser administradores o no. Un administrador puede crear más empleados, mientras que los no administradores únicamente tienen acceso a los pacientes.

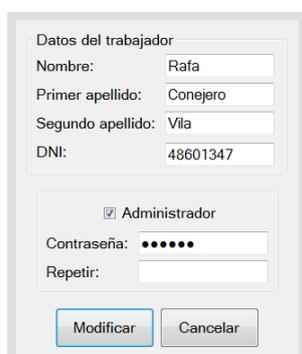
El formulario muestra los datos de un trabajador: Nombre: Rafa, Primer apellido: Conejero, Segundo apellido: Vila, DNI: 48601347. Incluye un checkbox para 'Administrador' que está marcado, y campos para 'Contraseña' y 'Repetir'. Al final hay botones para 'Modificar' y 'Cancelar'.

IMAGEN 43: FORMULARIO PARA CREAR UN TRABAJADOR.

Este formulario dispone de diversos cuadros de texto para introducir la información del empleado, además de un cuadro para introducir una contraseña e indicar con un “CheckBox” si es administrador o no.

#### 4.6.10. FORMULARIO “CONTRASEÑA”

Este formulario se puede acceder desde el formulario “Pacientes” en la sección “Paciente”.

Este será el empleado para los pacientes para modificar o crear una nueva contraseña. En el caso de que el paciente no recuerde su contraseña anterior y el fisioterapeuta acceda a sus datos, puede establecer un acceso predeterminado sin contraseña y posteriormente el paciente crear su nueva contraseña.

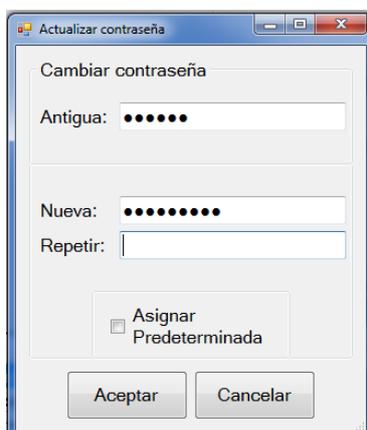
La ventana se titula 'Actualizar contraseña' y contiene el subtítulo 'Cambiar contraseña'. Incluye campos para 'Antigua', 'Nueva' y 'Repetir', todos con caracteres ocultos por puntos. Hay un checkbox para 'Asignar Predeterminada' que no está marcado. Al final hay botones para 'Aceptar' y 'Cancelar'.

IMAGEN 44: VENTANA PARA REALIZAR EL CAMBIO DE CONTRASEÑA A UN PACIENTE.

#### 4.6.11. CLASES INTERNAS

Las clases internas son un modelo que define métodos, estados y variables apropiados para trabajar mejor en la aplicación. El código completo se encuentra en el anexo 7.

Para poder gestionar los **pacientes**, las **sesiones**, los **ejercicios** y los **trabajadores**, se utilizan clases que definen un objeto que se utilizará en la aplicación.

Todos disponen de un constructor vacío y otro introduciendo todas las variables. Además, un método que vacía la información almacenada en el objeto representado por esa clase. En todos los casos excepto la clase “trabajadores”, contienen otra clase interna donde almacena menos información y con un orden distinto. Esto se debe a que a la hora de realizar una búsqueda de pacientes, sesiones y ejercicios, no requiere de toda la información para mostrar en el resultado. Hasta que no se seleccione uno de los resultados obtenidos, no se realiza la consulta para obtener toda su información. Por ejemplo, para realizar la búsqueda de pacientes, no se requiere el sexo, la contraseña o las fechas de inicio o finalización de la terapia. Son datos menos importantes.

Además la clase “**ejercicios virtuales**” contiene la información de los ejercicios base que pueden ser ejecutados en el entorno 3D. Únicamente contiene un identificador, el nombre y el número del ejercicio.

Las demás clases se utilizan para crear métodos de trabajo. MySQL es un servidor que recibe consultas y éste procesa y devuelve un resultado. Para ello, requiere de una clase **conectorMySQL**, encargada de realizar la conexión entre esta aplicación y la base de datos. En ella dispone de dos funciones: una para realizar la conexión con el servidor MySQL y otra para la desconexión.

Se utiliza una clase para controlar el **acceso a los formularios** y simplificar el código en los diferentes puntos que requieran dar acceso a un formulario. Este dispone de métodos para abrir, colocar y construir los formularios según requiera la aplicación.

Se dispone de una clase llamada “**operaciones**” con todos los métodos necesarios para comunicarse con el servidor MySQL y conversores necesarios para el correcto funcionamiento. En esta clase existen métodos para agregar, modificar y borrar pacientes, sesiones, ejercicios, trabajadores; para realizar búsquedas de estos mismos ya sea mediante palabras exactas o parciales, o con una o más variables de búsqueda. Otros métodos que se utilizan cuando después de realizar la búsqueda se selecciona que paciente, sesión, ejercicio o trabajador se desea abrir y obtener toda la información completa. Otro apartado dentro de esta clase son las funciones de subir y bajar el número de orden de las sesiones y ejercicios; los conversores de formato y comprobadores de fechas. MySQL dispone de unas variables “DateTime” donde, de una forma eficiente, almacena información temporal. El inconveniente es que su formato de almacenamiento no es el estándar conocido por personas y para ello hay que cambiarlo, ya sea para recibir la información de la base de datos o enviarla.

Además, dispone de otros métodos para crear nuevas tablas de sesiones y ejercicios. Y no olvidar las funciones para importar y exportar.

Por último, la clase que realiza **conexiones externas**, es la encargada en primer lugar, de recibir la información de configuración para determinar donde se almacenan los archivos adjuntos. Además, disponer de los métodos para adjuntar, visualizar y borrar los archivos a cada paciente.

---

# CAPÍTULO 5. CONCLUSIONES

En el TFG se ha desarrollado una aplicación que podría estar instalada en las consultas de los fisioterapeutas como ayuda para los pacientes que no requieran de una atención exhaustiva por parte del fisioterapeuta, y, por tanto, no ser necesaria su visualización directa de los ejercicios de rehabilitación. Además, aporta al especialista información de gran utilidad de los ejercicios específicos que se pueden realizar, como puede ser las repeticiones realizadas, el ángulo máximo alcanzado en la sesión, incluso la duración de cada ejercicio, entre otros.

A pesar de encontrarse con múltiples dificultades a lo largo de la ejecución del proyecto, todas se han podido subsanar y se han podido cumplir todos los objetivos planteados inicialmente.

Por ejemplo, una de las dificultades a destacar apareció a la hora de preparar el entorno tridimensional y crear la aplicación. Aparecieron problemas inesperados con el control del avatar. No obstante, se buscaron alternativas para proseguir con el desarrollo de la aplicación, como la utilización de una animación del avatar para representar el movimiento del usuario.

Tras la finalización de la realización del TFG, se realizaron pruebas satisfactorias con pacientes sanos, comprobando el correcto funcionamiento correcto de todos los elementos de la aplicación.

A título personal, realizar una aplicación de estas características, me ha resultado muy interesante, y me ha brindado la posibilidad de aprender mucho en múltiples campos en los que no tenía experiencia previa. No sólo por el entorno tridimensional controlado por Kinect V2, (utilizando Blender como herramienta gratuita, sin compatibilidad con este periférico), sino, también, por el hecho de enlazar diversos programas (en diversos lenguajes) a una base de datos, y realizar comunicaciones entre ellos. También me ha permitido aprender los siguientes lenguajes de programación: Python, C# y SQL.

Como trabajo futuro se pretende completar la aplicación, añadiendo un catálogo más extenso de ejercicios a realizar, incluso para rehabilitación de otras zonas del cuerpo, como son el cuello, tronco y extremidades inferiores. En el caso de disponer de varios puestos con la aplicación desarrollada en una misma consulta, se podría poner un único equipo como servidor de la base de datos y que el resto accedieran a él. Además, se podrían añadir funciones con representaciones gráficas de los datos obtenidos en las sesiones (ángulos máximos alcanzados, por ejemplo) para estudiar mejor la evolución de los pacientes y mejorar la experiencia del fisioterapeuta. En el ámbito interno de la aplicación, se podría crear una herramienta puente entre Kinect y Blender, y no depender del programa Ni Mate, además de mejorar la movilidad del avatar para sustituir las animaciones. Para mejorar la parte del paciente, también se podrían añadir videos demostrativos de los ejercicios, recomendaciones acústicas y botones de ayuda, entre otros aspectos.

De este modo, la aplicación conseguiría un mayor seguimiento de la rehabilitación del paciente, si éste tuviera la necesidad de tratar diversas zonas del cuerpo para recuperar su salud.

---

# CAPITULO 6. BIBLIOGRAFÍA

La mayoría de las fuentes encontradas son direcciones web de corporaciones, que se han indicado, la mayoría a pie de página en la secciones de la memoria del proyecto donde han sido consultadas.

Además de dichas referencias, se han consultado las siguientes fuentes:

[1-FISIO] Colegio de fisioterapeutas. Recuperado el 2017 de 06 de 22, de Colegio profesional de fisioterapeutas de madrid: <https://www.cfisiomad.org>

[2-WIKI] Recuperado el 19 de 06 de 2017, información obtenida de Wikipedia: [https://es.wikipedia.org/wiki/Interfaz\\_de\\_usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario)

[3-WIKI] Recuperado el 20 de 06 de 2017, de Wikipedia: <https://en.wikipedia.org/wiki/Kinect>

[4-KINECT] Recuperado el 2017 de 06 de 21, información obtenida de developer.microsoft.com: <https://developer.microsoft.com/es-es/windows/kinect/hardware-setup>

[5-ROQUE] Miguel Angel Roque. *Canal Youtube personal*. Recuperado el 2017 de 06 de 28, de <https://www.youtube.com/channel/UCzhdAv-HIN6lvHobILG1NDg>

[6-NUÑEZ] Ángela Núñez Castaín, (2003). Obtenido de la web del Ministerio de Educación, Cultura y Deporte: [http://recursostic.educacion.es/descartes/web/materiales\\_didacticos/Vectores3D\\_d3/vectores3D\\_15.htm](http://recursostic.educacion.es/descartes/web/materiales_didacticos/Vectores3D_d3/vectores3D_15.htm)