



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

Ingeniería de computadores

TRABAJO FIN DE GRADO

PhyRe UP! Sistema basado en realidad aumentada y
gamificación para la rehabilitación física

David Carneros Prado

Julio, 2020



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

Tecnologías y Sistemas de la Información

Ingeniería de computadores

TRABAJO FIN DE GRADO

**PhyRe UP! Sistema basado en realidad
aumentada y gamificación para la
rehabilitación física**

Autor: David Carneros Prado

Tutor: Dr. Javier Alonso Albusac Jiménez

Co-Tutor: Dr. David Vallejo Fernández

Julio, 2020

PhyRe UP! Sistema basado en realidad aumentada y gamificación para la rehabilitación física
© David Carneros Prado, 2020

Este documento se distribuye con licencia CC BY-NC-SA 4.0. El texto completo de la licencia puede obtenerse en <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

La copia y distribución de esta obra está permitida en todo el mundo, sin regalías y por cualquier medio, siempre que esta nota sea preservada. Se concede permiso para copiar y distribuir traducciones de este libro desde el español original a otro idioma, siempre que la traducción sea aprobada por el autor del libro y tanto el aviso de copyright como esta nota de permiso, sean preservados en todas las copias.

Este texto ha sido preparado con la plantilla \LaTeX de TFG para la UCLM publicada por [Jesús Salido](#) en GitHub¹ y Overleaf² como parte del curso « *\LaTeX esencial para preparación de TFG, Tesis y otros documentos académicos*» impartido en la Escuela Superior de Informática de la Universidad de Castilla-La Mancha.



¹https://github.com/JesusSalido/TFG_ESI_UCLM

²<https://www.overleaf.com/latex/templates/plantilla-de-tfg-escuela-superior-de-informatica-uclm/phjgscmfqtsw>

TRIBUNAL:

Presidente: _____

Vocal: _____

Secretario: _____

FECHA DE DEFENSA: _____

CALIFICACIÓN: _____

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

*A mis padres
y mis abuelos*

Resumen

La rehabilitación física de pacientes, particularmente los afectados por enfermedades neurológicas, supone comúnmente que el proceso de rehabilitación reúna a terapeuta y paciente en el mismo lugar físico. Esto plantea dificultades y limitaciones para el paciente, y habitualmente también para sus familiares, y representa un incremento en los costes económicos afrontados por el sistema de salud. Otro problema que plantea este contexto reside en que el tiempo que el rehabilitador puede estar con el paciente es limitado, y por tanto los ejercicios que el paciente no ha podido realizar en la consulta deberá realizarlos en su domicilio, lo que puede desencadenar desmotivación y, por consiguiente, el abandono de los mismos.

En este contexto, donde las herramientas de telemedicina aportan un valor añadido, el trabajo fin de grado que se plantea se centra en el diseño y desarrollo de un sistema de rehabilitación a distancia basado en realidad aumentada que aborde la problemática mencionada anteriormente y, simultáneamente, ofrezca mecanismos de interacción para que el paciente pueda llevar a cabo el proceso de rehabilitación de manera autónoma. Desde el punto de vista del terapeuta, el sistema posibilita la definición y el seguimiento de los ejercicios. El sistema además motivará al paciente mediante técnicas de gamificación para incentivarle a realizar los ejercicios con el fin de reducir el abandono de los mismos.

En el proceso del diseño y desarrollo del sistema será necesario estudiar las capacidades computacionales y las limitaciones de los dispositivos de realidad aumentada y *tracking* de articulaciones, eligiendo la configuración adecuada y abordando las posibilidades que estos ofrezcan en el contexto de proporcionar *feedback* visual al paciente en tiempo real, con el mínimo uso de componentes externos que dificulten la realización de los ejercicios.

Abstract

The physical rehabilitation of patients, particularly those affected by neurological diseases, commonly assumes that the rehabilitation process brings the therapist and the patient together in the same physical location. This poses difficulties and limitations for the patient, and usually for their relatives as well, and represents an increase in the economic costs faced by a health system. Another problem posed by this context is that the time that the rehabilitator can be with the patient is limited, and therefore the exercises that the patient has not been able to perform in the presence of the therapist must be performed at home, which can trigger demotivation and, consequently, the abandonment of said exercises.

In this context where telemedicine tools provide added value, the proposed final degree project would focus on designing and developing an augmented reality system that addresses the aforementioned problem and, simultaneously, offers mechanisms of interaction based on augmented reality so the patient can carry out the rehabilitation process autonomously. On the other hand, the definition and follow up of the exercises by the therapist is also contemplated. The system will motivate the patient through gamification techniques to encourage them to perform the exercises in order to reduce the abandonment of said exercises.

In the process of design and development of the prototype it will be necessary to study the computational capabilities and limitations of the augmented reality device, choosing the appropriate configuration and addressing the possibilities it offers in the context of providing real-time visual feedback to the patient.

AGRADECIMIENTOS

Me gustaría comenzar dando las gracias a mis padres, Carmelo y Celia, a mi hermana Celia y a Carlota porque gracias a su apoyo y ayuda constante, hasta en los peores momentos, he conseguido ser quien soy y llegar a donde he llegado.

Cabe destacar mi agradecimiento a mis compañeros de clase de ingeniería de computadores, con vosotros se formó algo más que una amistad, que gracias a esa risas y buenos momentos convirtieron las clases en un hogar para mí.

Gracias a todos los que han estado conmigo estos cuatro años, pero sobretodo he de agradecer la compañía de Mario, Andrés y Sergio por todos los momentos juntos y su apoyo. Sin vosotros nada hubiera sido igual.

Por último, he de agradecer a mi director Javier por sus incontables muestras de apoyo, de cariño y su gran labor docente me han servido de inspiración en todo momento. Gracias también a mi co-director David por su ayuda y apoyo siempre que lo necesitaba. Doy las gracias además a FK por haberme podido dejar los dispositivos necesarios para el desarrollo de este proyecto.

David Carneros Prado
Ciudad Real, 2020

LISTADO DE ACRÓNIMOS

<i>TFG</i>	: Trabajo de fin de grado.
<i>AVD</i>	: Actividad de vida diaria.
<i>EESE</i>	: Encuesta Europea de Salud en España
<i>OMS</i>	: Organización Mundial de la Salud.
<i>RA</i>	: Realidad aumentada.
<i>RV</i>	: Realidad virtual.
<i>AF</i>	: Actividad física.
<i>SEN</i>	: Sociedad Española de Neurología.
<i>CUDA</i>	: Arquitectura Unificada de Dispositivos de Cómputo.
<i>HMD</i>	: <i>Head-mounted display</i> .
<i>GGV</i>	: Mirada, gestos y voz.
<i>ToF</i>	: Tiempo de vuelo.
<i>GPU</i>	: Unidad de procesamiento gráfico.
<i>HPU</i>	: Unidad de procesamiento holográfico.
<i>IMU</i>	: Unidad de medición inercial
<i>SoC</i>	: Sistema en chip
<i>DSP</i>	: Procesador digital de señales
<i>RAM</i>	: Memoria de acceso aleatorio
<i>SRAM</i>	: Memoria estática de acceso aleatorio.
<i>SDK</i>	: Kit de desarrollo <i>software</i> .
<i>FPS</i>	: Fotogramas por segundo.
<i>ORM</i>	: Mapeo objeto-relacional.
<i>SGDB</i>	: Sistema gestor de base de datos.

ÍNDICE GENERAL

Resumen	IX
Abstract	XI
Agradecimientos	XIII
Listado de acrónimos	XV
Índice de figuras	XIX
Índice de tablas	XXI
Índice de listados	XXIII
Índice de algoritmos	XXV
1. Introducción	1
1.1. Estructura del documento	5
2. Objetivos	7
2.1. Objetivo general	7
2.2. Objetivos específicos	7
3. Estado del arte	11
3.1. Juegos serios	11
3.1.1. Taxonomía de Sawyer y Smith	12
3.2. Exergames	13
3.2.1. Antecedentes	15
3.2.2. ¿Realidad virtual o realidad aumentada?	16
3.3. Gamificación	19
3.4. Estudio y elección de entornos de RA y dispositivos hardware	20
3.4.1. Hololens	22
3.4.2. Dispositivo para el seguimiento de las extremidades	23
4. Metodología	27
4.1. Metodología de trabajo	27
4.2. Recursos	28
4.2.1. Recursos hardware	28
4.2.2. Recursos software	29

5. Arquitectura	33
5.1. Visión general	33
5.2. Aplicación de Tracking del cuerpo	34
5.2.1. Visión general	34
5.2.2. Inicio y acceso al dispositivo	35
5.2.3. Obtención de la posición de las articulaciones	37
5.2.4. Comunicación con la aplicación de RA	39
5.3. Aplicación RA para el terapeuta	39
5.3.1. Entradas	40
5.3.2. Tracking de manos	41
5.3.3. Definición de ejercicios	44
5.3.4. Definición y asignación de rutinas	46
5.4. Aplicación de RA para la rehabilitación del paciente	46
5.4.1. Representación del ejercicio de rehabilitación	47
5.4.2. Realización de los ejercicios	47
5.5. Gamificación	49
5.6. Backend	50
5.6.1. Vista general	50
5.6.2. Base de datos	50
5.6.3. Contenedores	53
5.6.4. API Rest	54
6. Resultados	57
6.1. Physical Rehabilitation UP!	57
6.1.1. Tracking de manos	58
6.1.2. Aplicación del terapeuta	58
6.1.3. Aplicación del paciente	63
6.1.4. Backend	65
6.2. Distribución del trabajo	66
6.2.1. Paquetes de trabajo	66
6.2.2. Iteraciones	70
6.3. Costes de desarrollo	72
6.4. Experimentación	73
7. Conclusiones	77
7.1. Objetivos alcanzados	78
7.2. Justificación de competencias adquiridas	79
7.3. Trabajo futuro	80
Bibliografía	81
A. Fotografía del sistema	87
B. Experimentación	89

ÍNDICE DE FIGURAS

3.1.	Comparación entre realidad virtual y realidad aumentada	15
3.2.	Sistema vestibular humano	17
3.3.	Ejemplo de realidad aumentada en el móvil con la aplicación <i>10.000 moving cities</i> (Marc Lee / CC 4.0)	21
3.4.	Dispositivo <i>Microsoft HoloLens</i> By Ramadhanakbr - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=52546267	22
3.5.	Ejemplo de uso de las <i>HoloLens</i> para la educación. Imagen por Hoshinim / CC BY-SA (https://creativecommons.org/licenses/by-sa/4.0)	24
3.6.	Dispositivo <i>kinect</i> para la consola <i>Xbox 360</i>	24
3.7.	Azure kinect DK	25
3.8.	Articulaciones obtenidas por el dispositivo Azure Kinect. Documentación de Microsoft https://docs.microsoft.com/en-us/azure/kinect-dk/body-joints	26
4.1.	Esquema de desarrollo iterativo e incremental	28
5.1.	Esquema de los sistemas que componen <i>PhyRe UP!</i>	33
5.2.	Esquema de la aplicación de <i>tracking</i> del esqueleto	35
5.3.	Articulaciones obtenidas por el dispositivo Azure Kinect. Documentación de Microsoft	38
5.4.	Vista general de la aplicación de RA para el terapeuta	40
5.5.	Ejes de coordenadas para la aplicación de realidad aumentada y para la de <i>tracking</i> del cuerpo	41
5.6.	Marca para obtener la posición y rotación del dispositivo de <i>tracking</i>	42
5.7.	Creación del eje de coordenadas para el dispositivo de <i>tracking</i>	43
5.8.	Vista general del <i>trackeo</i> de manos	43
5.9.	Posición del punto guardado para el ejercicio	45
5.10.	Vista general de la aplicación de RA para el paciente	46
5.11.	Vista general del backend	50
5.12.	Esquema de la base de datos	51
5.13.	Comparación entre <i>Docker</i> y máquinas virtuales	53
5.14.	Comunicación entre los contenedores y el exterior	55
6.1.	Logotipo del sistema <i>PhyRe UP!</i>	57
6.2.	<i>Tracking</i> de mano vista desde la aplicación de RA	58
6.3.	Menú principal de la aplicación RA para el terapeuta	59
6.4.	<i>Tracking</i> de mano vista desde la aplicación de RA	59
6.5.	Captura de la grabación de un ejercicio	60
6.6.	Últimas fases de la definición de ejercicios	60

6.7. Entrada de texto para las aplicaciones de RA	61
6.8. Creación de rutinas	61
6.9. Asignación de rutinas	62
6.10. Menú para elegir el ejercicio para ver	62
6.11. Visualización de ejercicio	63
6.12. Primeras ventanas de la aplicación de RA del paciente	63
6.13. Menú de selección de ejercicios a realizar	64
6.14. Mensajes de información mostradas antes de la realización del ejercicio	64
6.15. Vista del ejercicio antes de ser realizado por el paciente	65
6.16. Paciente realizando un ejercicio	65
6.17. Topología del <i>backend</i> desde la vista de <i>Microsoft Azure</i>	66
6.18. Diagrama de Gantt que representa la planificación del proyecto	69
6.19. Ejercicio definido para las pruebas	74
6.20. Resultado del ejercicio sin <i>feedback</i> . En azul el ejercicio definido por el terapeuta y en rojo las trayectorias trazadas por los sujetos de prueba	74
6.21. Trayectorias trazadas por los sujetos del experimento haciendo uso de la aplicación de RA. En azul el ejercicio definido por el terapeuta y en verde las trayectorias trazadas por los sujetos de prueba	75
6.22. Comparación del resultado con y sin <i>feedback</i> . En azul el ejercicio definido por el terapeuta, en rojo las trayectorias trazadas por los sujetos de prueba sin <i>feedback</i> y en verde con <i>feedback</i>	76
A.1. Fotografía del sistema <i>PhyRe UP!</i> con anotaciones	87
A.2. Fotografía del sistema <i>PhyRe UP!</i> sin anotaciones	88
B.1. Ejercicio visto desde la Aplicación de RA	89
B.2. Resultados de la experimentación con sujetos de prueba. Rojo sin <i>feedback</i> , verde con <i>feedback</i> y azul el ejercicio definido	92

ÍNDICE DE TABLAS

1.1. Tabla de la Encuesta Europea de Salud en España table	2
3.1. Tabla de clasificación de videojuegos serios según Sawyer y Smith (2008) .	13
5.1. Diferentes modos de funcionamiento de la cámara de profundidad del <i>Azure kinect</i>	35
5.2. Tabla con resultados de comparación con diferentes configuraciones del sistema	37
5.3. Acciones y eventos desencadenados	49
6.1. Resumen de la iteración 1	70
6.2. Resumen de la iteración 2	70
6.3. Resumen de la iteración 3	71
6.4. Resumen de la iteración 4	71
6.5. Resumen de la iteración 5	71
6.6. Resumen de la iteración 6	72
6.7. Resumen de la iteración 7	72
6.8. Coste total del proyecto	73

ÍNDICE DE LISTADOS

5.1.	Estructura usada para almacenar los parámetros de configuración	36
5.2.	Hilo de obtención de resultados	38
5.3.	Ejemplo de la entidad de <i>RoutineResult</i>	52
5.4.	<i>Dockerfile</i> para la creación de la imagen del servidor de la API Rest	54
5.5.	Ejemplo simplificado del controlador de <i>Routine</i>	56
5.6.	Ejemplo simplificado del servicio de <i>Routine</i>	56
B.1.	Ejemplo de archivo csv con la trayectoria del paciente	90
B.2.	<i>Script</i> para dibujar los resultados de la experimentación	91

ÍNDICE DE ALGORITMOS

5.1. Algoritmo de definición de ejercicios	44
5.2. Algoritmo de realización de ejercicios de rehabilitación	48

CAPÍTULO 1

INTRODUCCIÓN

La especialidad médica de rehabilitación, como la conocemos actualmente, surge en Estados Unidos a comienzos del siglo XX, con la figura del médico Dr. Frank Kursten. Cuando, debido a un cuadro de tuberculosis que contrajo, pudo darse cuenta que los períodos de recaída de la enfermedad de los diferentes pacientes en el centro médico se relacionaban con el desacondicionamiento físico [19]. Siendo ahora, la rehabilitación física, un aspecto clave en la medicina moderna ya que es necesaria, por ejemplo, para la recuperación tras una operación, lesiones físicas o para diversas enfermedades neurológicas.

Ahondado un poco más en el aspecto concreto que nos ocupa, a la hora de valorar la importancia de la rehabilitación en distintas enfermedades, tanto de tipo físico como neurológica, nos puede ilustrar como ejemplo, los siguientes datos. Unos 179 millones de europeos viven con un trastorno cerebral. De hecho, se estima que una de cada tres personas sufre o sufrirá algún trastorno neurológico o psiquiátrico. El coste global para los presupuestos sanitarios nacionales europeos se estima en 800.000 millones de euros al año, lo que equivale a cinco veces el presupuesto de la Unión Europea. Según el *European Brain Council*¹, esto equivale a 1,5 millones de euros por minuto [32].

Un ejemplo de enfermedad neurológica es el caso del ictus. En España se producen casi 120.000 casos de ictus al año, de los cuales, la mitad quedan con secuelas discapacitantes o fallecen y, aunque su mortalidad se ha reducido con los avances de la medicina, se espera que en los próximos 25 años su incidencia aumente en un 27 % según los datos de la SEN (Sociedad Española de neurología)². Cada paciente ingresado en las unidades de ictus tiene un coste de 27.711 euros al año para el Estado [1].

Una de las discapacidades comunes que sufren las personas afectadas por enfermedades neurológicas es la pérdida brusca de fuerza, movilidad o sensibilidad en partes del cuerpo, siendo necesaria una rehabilitación física que puede llegar a ser muy duradera o llegar a continuar durante toda la vida del paciente. Por consiguiente, el tiempo es un factor crucial

¹<https://www.braincouncil.eu>

²<http://www.sen.es>

en este tipo de rehabilitaciones ya que cuanto antes se traten estas discapacidades habrá una mayor probabilidad de que el paciente pueda recuperar, en cierta parte, el grado de movilidad anterior a la enfermedad. Esto se puede observar en el estudio realizado por A. Maulden [22] con pacientes que habían sufrido un ictus moderado o grave en el que puso de manifiesto que el inicio temprano de la rehabilitación se asocia, de forma significativa, con una mejor evolución del estado de la lesión, alcanzando mayores puntuaciones en la escala de actividades de vida diaria (AVD). Al contrario que los pacientes que empezaron con retraso la rehabilitación que fueron asociados con una peor evolución funcional.

Debido al aumento de esperanza de vida y al envejecimiento de la población se está produciendo un aumento en el número de personas que realizan una visita al médico. Según la última encuesta Europea de salud en España (ESEE), publicada por el Ministerio de Sanidad³, se muestra como en el último año de la encuesta (2014), la población de 15 años o más que había tenido una consulta con el médico ese año era de un 84,49 %, lo que significa un aumento de casi un 20 % con respecto al año 1987 donde el porcentaje se situaba en 68,3 %. Se puede ver en la Tabla 1.1 como han ido aumentando las visitas al médico desde 1987 al último año de la ESEE (2014). En la encuesta nacional de salud realizada en 2017 destaca otro aumento en cuanto al porcentaje de personas que acuden al médico situándose en un 86.8 % (91,4 % mujeres, 82,1 % hombres) [26]. Lo que se traduce en que el tiempo que los profesionales de la salud pueden dedicar a cada paciente se está reduciendo al aumentar la cantidad de pacientes a tratar, que a su vez también produce un incremento en el gasto en salud al necesitar más recursos humanos para poder cubrir con todas a todos los pacientes.

Tabla 1.1: Consulta al médico en el último año según sexo y año de encuesta 1987-2014. Mas la encuesta [26] (2017) (%)

	1987	1993	1995	1997	2001	2003	2006	2011	2014	2017
Total	68.33	67.75	72.61	72.76	74.93	82.38	82.93	81.07	84.49	86.8
Hombres	63.04	62.57	66.96	67.69	68.54	76.39	76.36	75.85	78.59	82.1
Mujeres	73.02	72.51	77.81	77.43	80.74	88.11	89.26	86.04	90.11	91.4

Actualmente en las pequeñas poblaciones de la península se está produciendo una pérdida de los centros sanitarios que están centralizándose en las capitales. Lo que es un gran problema para los habitantes de estos municipios ya que en caso de necesitar atención médica deberán desplazarse hasta el centro sanitario que puede estar a varios kilómetros de distancia. Siendo esto también un problema para el hospital por qué, en el caso de que la persona no pueda desplazarse al hospital, se deberá enviar un vehículo sanitario a recoger al paciente o los médicos deberán acudir al domicilio de este paciente. Lo que conlleva para el hospital un gran coste de capital al tener que disponer de más sanitarios para realizar estas labores.

³<https://www.msbs.gob.es>

Se ha demostrado que la terapia activa es más eficaz en este tipo de rehabilitaciones [27] ya que llevar a cabo ejercicios de manera repetitiva ayuda a superar las dificultades motoras que los pacientes poseen. Cuando el paciente realiza esta rehabilitación, realiza los ejercicios junto a su terapeuta y, es este profesional el que está pendiente de como el paciente esta realizado los ejercicios. El terapeuta se encarga de guiar al paciente para que realice de forma correcta los ejercicios. Además, el profesional podrá comprobar si los ejercicios que esta realizando el paciente son demasiado avanzados para su grado de lesión o si por el contrario necesita ejercicios más acorde a sus exigencias. El problema reside en que el tiempo que el rehabilitador puede estar con el paciente es limitado, y por tanto, la cantidad de ejercicios que pueden realizar en consulta no es el adecuado.

Los ejercicios que los pacientes no han realizado en consulta deberán ser realizados en casa y seguir con una rutina hasta la siguiente consulta en la que el personal sanitario encargado de la rehabilitación pueda comprobar el avance. El hecho de que el paciente tenga que desplazarse hasta la consulta puede suponer un riesgo si su estado de salud es delicado y, en caso contrario, que el terapeuta tenga que desplazarse hasta la casa del paciente supone un elevado coste de tiempo y a su vez, de dinero.

Por otro lado, no solo tener que desplazarse es un problema. El hecho de que el paciente tenga que realizar los ejercicios en casa, ejercicios que son repetitivos, sin la motivación de un profesional guiándole a hacer los ejercicios puede desencadenar una falta de compromiso por parte del paciente [20]. Dicha falta de motivación llevará al paciente a abandonar los ejercicios realizándolos solo cuando vaya a consulta.

De igual forma, no solo la falta de motivación puede ser un problema. El paciente debe realizar de forma correcta los ejercicios mandados por el terapeuta, ya que una realización errónea de los ejercicios puede provocar que el paciente no mejore o incluso que se agrave su condición. Según [41] tan solo un 31 % de las personas sometidas a terapias de rehabilitación realiza los ejercicios de forma correcta en sus domicilios.

Como solución a los problemas planteados anteriormente surge la telemedicina. La telemedicina puede parecer un concepto de la época presente, pero es una idea que proviene desde hace unas décadas. A principios del siglo XIX se puede ver la primera aparición del concepto de telemedicina en un artículo de la revista «*Radio News*» titulado *Doctor por radio*, en el cual se mostraba la idea de telemedicina incluyéndose una imagen de una máquina con televisor y radio que permitía la comunicación del paciente con el médico a distancia.

La palabra telemedicina proviene del griego «*τελε*» (tele) que significa distancia, y «*medicina*» que proviene del latín, deriva de *medicus* (médico), palabra que a su vez deriva de *mederi* (cuidar, curar, tratar). Pero la definición de telemedicina no es sencilla. A simple vista se puede deducir que telemedicina significa medicina a distancia (diagnostico, tratamiento, etc) haciendo uso de los recursos tecnológicos para eliminar las barreras como la distancia.

De ahí obtenemos la definición de la OMS del concepto de telemedicina: “el suministro de servicios de atención sanitaria en los que la distancia constituye un factor crítico, realizado por profesionales que apelan a tecnologías de la información y de la comunicación con objeto de intercambiar datos para hacer diagnósticos, preconizar tratamientos y prevenir enfermedades y heridas, así como para la formación permanente de los profesionales de atención de salud y en actividades de investigación y evaluación, con el fin de mejorar la salud de las personas y de las comunidades en que viven” [33].

En nuestros días, el avance de la medicina y de las tecnologías de la información y de la comunicación han dado como resultado que la idea de telemedicina se haga real. La telemedicina con el uso respaldado por parte de los profesionales de la salud, además del evidente beneficio para el paciente, al no tener que desplazarse hasta el centro sanitario, supone una solución para los lugares cuyo acceso es complicado que podrán recibir atención médica sin necesidad de desplazarse. De igual manera supone un gran beneficio para los sanitarios ya que a través de estos sistemas pueden dotar a los pacientes de una asistencia de mayor duración y calidad.

Algunas de estas tecnologías son la realidad virtual (RV) y la realidad aumentada (RA) que proporcionan mecanismos avanzados de representación. La realidad virtual es un entorno de escenas con apariencia real que son contempladas por el usuario a través de un dispositivo conocido como gafas o casco de realidad virtual. En la realidad aumentada sin embargo el usuario visualiza el mundo real a través de un dispositivo con información gráfica añadida en él. Es decir, el usuario ve el mundo real con información superpuesta. En los últimos años la RA y RV se han aplicado de diversas formas en la telemedicina; desde aplicaciones para mejorar el aprendizaje de esta profesión hasta aplicaciones para la rehabilitación. Estas tecnologías suponen un gran avance en el contexto de la rehabilitación, ya que con su uso los pacientes pueden ser guiados en los ejercicios y el terapeuta puede hacer un seguimiento de él/ella, asegurándose de esta manera que el paciente está realizando los ejercicios de forma correcta, lo que reduce la cantidad de pacientes que realizan los ejercicios de forma errónea.

Como se ha comentado anteriormente, la realización de los ejercicios de rehabilitación por parte de los pacientes, ejercicios que son monótonos y sin la ayuda de un terapeuta, pueden desencadenar una falta de motivación que a su vez derive en una recuperación más lenta o en el abandono de la rehabilitación. Una forma de solucionar estos problemas de falta de motivación es hacer uso de técnicas de gamificación. Estas técnicas presentan al paciente los ejercicios en forma de juegos que dotan de buen *feedback* al paciente y recompensan la realización de estos ejercicios. De esta manera, el paciente se mantiene motivado en la rehabilitación reduciendo así la tasa de abandono de los ejercicios.

Por todo lo expuesto hasta este momento, este TFG tiene como objetivo la propuesta de soluciones a las problemáticas expuestas presentando un sistema de telemedicina que ayude en la rehabilitación física de pacientes afectados por causas como bien puedan ser las rehabilitación tras una operación o la rehabilitación de enfermedades neurológicas. Para ello se quiere hacer uso de la realidad aumentada para posibilitar la autonomía del paciente y ayudarle en la realización de los ejercicios de rehabilitación de una forma guiada haciendo, que de esta forma, realice los ejercicios de forma correcta y por tanto pueda avanzar con su recuperación. La realidad aumentada supondrá una mejora frente a los sistemas actuales, como es el caso de los sistemas donde el *feedback* que recibe el paciente es a través de un monitor, al proporcionar un mejor *feedback* tridimensional al paciente. Asimismo se pretende conseguir, con esta autonomía a la hora de realizar los ejercicios, que el terapeuta pueda brindar una atención de mayor tiempo y calidad a los pacientes afectados por este tipo de enfermedades o lesiones. Del mismo modo, se propone que el terapeuta pueda tener control de los avances del paciente en cuanto a la realización de los ejercicios y de esta forma poder asignar los ejercicios que mejor concuerden con su progresión y grado de lesión.

Otra de las problemáticas a solucionar es la falta de motivación del paciente, ya que como se ha comentado es un factor clave en los abandonos de los ejercicios de rehabilitación. Para dar una solución a estos problemas se pretende usar técnicas de gamificación al sistema de realidad aumentada para convertir estos ejercicios en pequeños «juegos». De esta forma se pretende motivar y premiar al paciente cuando realice los ejercicios. Además, de esta forma el paciente se sentirá recompensado cuando realice los ejercicios, lo que se traducirá a una mayor sensación de progreso.

1.1. ESTRUCTURA DEL DOCUMENTO

En esta sección se explicará la estructura del documento que cubrirá todo el proceso de diseño y desarrollo del TFG.

- *Capítulo 1: Introducción*

Capítulo actual del documento. En este se aborda el contexto del TFG y la problemática existente a la que se pretende dar solución.

- *Capítulo 2: Objetivos*

En este capítulo se muestra de forma detallada cuál es el objetivo principal que se pretende conseguir con este proyecto y se divide en objetivos más específicos para poder completar el general.

- *Capítulo 3: Estado del arte*

Todos los proyectos tienen una base en la que se apoyan para poder lograr el objetivo propuesto. En este capítulo se va a realizar un estudio sobre los diferentes entornos

de RA frente a RV debido a los efectos colaterales de la total inmersión que pueden desencadenar en el paciente. Se estudiará la forma de aumentar la motivación del paciente usando técnicas de gamificación. Así como la elección del o los dispositivos que mejor se adapten a las necesidades estudiando las capacidades computacionales y las limitaciones de los mismos para elegir la configuración adecuada para el proyecto.

- *Capítulo 4: Metodología*

En este capítulo se expone la metodología utilizada durante el desarrollo del proyecto. Además, también se tratarán las herramientas y los recursos usados para llevarlo a cabo.

- *Capítulo 5: Arquitectura*

Este capítulo se centra en la descripción técnica del sistema, explicando los problemas que han surgido durante el desarrollo explicando el porqué de la solución elegida en cada momento y porqué es mejor que otras soluciones.

- *Capítulo 6: Resultados*

El capítulo de resultados tiene como objetivo exponer la aplicación de la metodología para obtener los resultados, así como los resultados obtenidos del sistema una vez terminado el desarrollo y el coste del sistema.

- *Capítulo 7: Conclusión*

Capítulo en el que se realizará una discusión sobre los resultados obtenidos, evaluando si se ha cumplido con los objetivos propuestos, justificando las competencias adquiridas y estableciendo potenciales líneas de trabajo futuro.

CAPÍTULO 2

OBJETIVOS

En este capítulo se pretende llevar a cabo una descripción detallada del objetivo que este proyecto pretende lograr, desarrollando, en primer lugar, el objetivo general y luego describiendo en detalle sus objetivos específicos.

2.1. OBJETIVO GENERAL

El objetivo general del proyecto es *diseñar y desarrollar un sistema remoto basado en realidad aumentada para la rehabilitación física que hará uso de técnicas de gamificación*. De esta manera se pretende hacer uso de la realidad aumentada para poder guiar a los pacientes que realizan estos ejercicios desde casa y que los realicen de forma correcta. Además, haciendo uso de técnicas de gamificación se pretende motivar al paciente para que realice los ejercicios y que no le parezcan monótonos y, por tanto, reducir la tasa de abandono de los ejercicios.

También se busca conseguir que el terapeuta pueda llevar un seguimiento de los avances del paciente sin tener que realizar el seguimiento de forma presencial, realizándose una evaluación del grado de mejora del paciente.

2.2. OBJETIVOS ESPECÍFICOS

Partiendo de la base del objetivo general definido anteriormente se pretende identificar objetivos más específicos que una vez completados nos permitan llegar al objetivo general. Los objetivos específicos son los siguientes:

1. **Estudio y evaluación de dispositivos *hardware*.**

Para realizar este proyecto será necesario estudiar las capacidades computacionales y las limitaciones de diferentes dispositivos *hardware* de realidad aumentada del mercado para, de esta forma, poder elegir el que mejor se adapte a la propuesta. La elección del dispositivo debe abordar la posibilidad que mejor *feedback* proporcione al paciente en tiempo real y elimine en todo lo posible los efectos adversos del entorno virtual.

También será necesario evaluar si es necesario algún dispositivo que permita hacer *tracking* del cuerpo para un seguimiento de la posición de las articulaciones en el espacio tridimensional. Esta información de las articulaciones será necesario para que el paciente realice el ejercicio. Un aspecto a destacar es que dicho dispositivo deberá obtener el *tracking* del cuerpo sin hacer uso de marcas o sensores para evitar la sobrecarga en pacientes con dificultades cognitivas o motoras.

2. Diseño y desarrollo de esquemas de interacción basados en realidad aumentada atendiendo a las capacidades computacionales del dispositivo escogido para el proyecto.

Este objetivo pretende encontrar el mejor esquema de interacción con el entorno virtual para que, desde la vista del paciente, usar el sistema no le suponga una molestia. Para ello se busca una interacción con el sistema lo menos intrusiva posible, es decir, que el número de controladores físicos que tenga que llevar el paciente sea el menor posible.

3. Desarrollo de un módulo *software* que permita al terapeuta definir los ejercicios.

Diseño y desarrollo de un un módulo *software* con el que el profesional sanitario defina los ejercicios para los pacientes. Además de la definición de ejercicios también podrá definir rutinas de ejercicios que asignará a los pacientes, indicando parámetros como la cantidad de repeticiones que debe realizar o el tiempo máximo de duración del ejercicio.

Estos ejercicios luego serán asignados a los pacientes que tendrán que realizarlos siguiendo las indicaciones del terapeuta.

4. Desarrollo de un *backend* que permita almacenar los ejercicios y los resultados de los mismos.

Una pieza fundamental en esta aplicación de telemedicina es que el terapeuta pueda seguir los progresos del paciente. Por tanto, es necesario poder almacenar los resultados de los ejercicios, estos resultados deben ser luego analizados y comparados para poder saber cómo está siendo la progresión del paciente. Lo que va a permitir al terapeuta poder conocer si el paciente está superando sus dificultades motoras sin necesidad de tener que realizar una consulta en el centro hospitalario.

5. Desarrollo de un módulo *software* basado en gamificación orientado a motivar a los pacientes para la realización de los ejercicios.

Como se ha comentado anteriormente la pérdida de motivación de los pacientes cuan-

do realizan los ejercicios de rehabilitación es un aspecto clave en el abandono de los mismos. Si el paciente no realiza los ejercicios puede no llegar a superar su grado de lesión o que el periodo de rehabilitación sea más duradero.

Por ello, gracias al uso de técnicas de gamificación se pretende captar la atención del paciente y que éste se sienta más motivado para realizar los ejercicios y de esta forma reducir el abandono induciendo en el paciente una mayor sensación de progreso. Igualmente se hará uso de técnicas de gamificación para poder guiar al paciente en la realización del ejercicio para conseguir una correcta ejecución del mismo.

6. Desarrollo de un módulo *software* que permita evaluar automáticamente los ejercicios de rehabilitación realizados por un paciente

Un pilar fundamental de este proyecto es la evaluación automática de los ejercicios realizados por un paciente para conocer su progreso. Con la evaluación del progreso se podrá conocer si el paciente está mejorando su grado de discapacidad, o si por el contrario no se está produciendo una mejora y, por tanto, los ejercicios recomendados por el terapeuta no han sido los adecuados para el grado de lesión.

Por este motivo es importante el diseño y desarrollo de un módulo *software* que sea capaz de analizar los resultados de los ejercicios y comunicar al personal sanitario, encargado de la rehabilitación, el estado del paciente sin necesidad de una evaluación presencial.

CAPÍTULO 3

ESTADO DEL ARTE

Todos los proyectos cuentan con una base teórica en la que se apoyan. En este capítulo se van a presentar los conceptos teóricos que van a ser decisivos para el desarrollo del proyecto y que van a ser necesarios para un completo entendimiento las decisiones tomadas durante el desarrollo. Primero se va a tratar la *taxonomía de Sawyer* sobre los juegos de salud. Tras haber explicado este concepto se va a introducir el concepto de «*exergames*». Una vez conocido este concepto, se va a comentar las pautas o recomendaciones que deben seguir los *exergames* para ser seguros y adecuados en función de la patología del paciente, además de introducir el concepto de gamificación. Una vez comentado el diseño de los *exergames* se va a explorar la efectividad de los entornos de realidad aumentada frente a la realidad virtual debido a los efectos colaterales de las distintas tecnologías. Por último, para concluir se va a realizar un estudio de los diferentes dispositivos del mercado para poder elegir la mejor configuración que se adapte a este proyecto.

3.1. JUEGOS SERIOS

Desde nuestro nacimiento y en las distintas fases de nuestra vida estamos en contacto con juegos o actividades lúdicas. En edades tempranas, el niño se relaciona con los juegos de forma continua, los juegos no solo aportan un entretenimiento para el niño si no que, según *Piaget e Inhelder* [34], los juegos con los que se relaciona el niño hasta los dos años de edad contribuyen de forma significativa a su desarrollo sensorial, aumentando la comprensión del mundo, la capacidad de superación y su interacción social. Durante la siguiente etapa de su crecimiento sigue haciendo uso de los juegos que le ayudarán a comprender la sociedad, fomentar la imaginación y creatividad, así como desarrollar sus habilidades comunicativas.

Aunque la mayoría de los efectos que se reportan de los juegos, sobre todo en la prensa cotidiana, son las consecuencias y riesgos negativos de jugar a videojuegos. Pero se puede ver en diversos estudios como los juegos bien enfocados presentan grandes beneficios. Según *Griffiths* [15] jugar a videojuegos produce una reducción en los tiempos de reacción del individuo además de una mejor coordinación visión-motora. Por lo cual, los beneficios de

los juegos pueden ser usados, en el ámbito de la educación, como mecanismo para practicar o aprender algunas habilidades siendo también el caso de las personas con necesidades especiales. Y, por supuesto, pueden ser usados para la salud y la rehabilitación. Debido a todo lo expuesto anteriormente nace el concepto de *juego serio*.

Un juego serio es un juego especialmente diseñado para producir un cambio en el jugador. Este cambio puede ser en el conocimiento, la capacidad física o cognitiva, la salud o el bienestar mental. Esta es la distinción del juego serio frente a un juego tradicional, cuyo fin es el mero entretenimiento [23]. Una de las primeras definiciones de juegos serios es dada por Zyda [51]: “una prueba mental, jugado con una computadora de acuerdo con reglas específicas, que utiliza el entretenimiento para promover la capacitación gubernamental, corporativa, educación, salud, políticas públicas y objetivos de comunicación estratégica”. Siendo ahora posible, dado el avance de las tecnologías de realidad aumentada y realidad virtual, realizar juegos serios enfocados en realizar actividad física (AF). Un concepto que va muy ligado al concepto de juego (o videojuego) serio, es el concepto de gamificación que será definido más adelante.

3.1.1. Taxonomía de Sawyer y Smith

La tarea de realizar una clasificación de los juegos serios no es, ni ha sido nada fácil. Para llevar a cabo esta tarea han surgido diferentes ideas de taxonomías de videojuegos serios, pero en este caso vamos a centrarlos por la taxonomía dada por *Sawyer y Smith* [38] en la que propone una tabla que clasifica los videojuegos serios dependiendo del dominio de aplicación y el organismo. Esto se puede observar en la Tabla 3.1.

En la primera columna de la Tabla 3.1 se muestran los organismos de destino del juego serio. *Sawyer y Smith* proponen siete organismos de destino. Estos organismos son: Gobiernos y ONGs, defensa, sistemas de salud, marketing y comunicaciones, educación, empresas e industria.

En la primera fila de la tabla se muestra el dominio de aplicación. Se proponen siete dominios de aplicación: juegos para la salud, publicitarios, para la formación, para la educación, la ciencia y la investigación, la producción y el empleo. Siendo la intersección entre filas y columnas el propósito del videojuego. Los dominios de aplicación que más destacan en las investigaciones y desarrollos de juegos serios en cuanto al número de publicaciones son los juegos para la salud, seguidos de los juegos publicitarios, para la formación y para la educación, siendo los organismos de destino más comunes los sistemas de salud, educación y empresas.

Si se presta especial atención a la categoría de sistemas de salud, se puede apreciar que sus dominios de aplicación van desde juegos para fomentar ámbitos de vida saludable, pasando por juegos de formación para los propios estudiantes de los diferentes campos de la salud,

Tabla 3.1: Tabla de clasificación de videojuegos serios según Sawyer y Smith (2008)

	Juegos para la Salud	Juegos Publicitarios	Juegos para la Formación	Juegos para la Educación	Juegos para la Ciencia y la Investigación	Producción	Juegos como Empleo
Gobiernos y ONGs	Educación para la salud y Respuesta a problemas de salud masivos	Juegos políticos (Campañas de partidos políticos)	Formación de empleados	Información pública	Recogida de datos / Planificación	Planificación de políticas y Estrategias.	Diplomacia / Estudios de opinión
Defensa	Rehabilitación y bienestar psicológico	Reclutamiento y propaganda	Formación de apoyo a los soldados	Educación en la escuela y en el hogar	Juegos de guerra / planificación	Planificación de la guerra e investigación de armamento	Mando y control
Sistemas de salud	Ciberterapia / Exergaming	Política de salud pública y Campañas de concienciación social	Juegos formativos para profesionales de la salud	Juegos para la educación de los pacientes y para la gestión de la enfermedad	Visualización y Epidemiología	Diseño y fabricación de biotecnología	Planificación y Logística de planes de salud publica
Marketing y Comunicaciones	Publicidad de tratamientos médicos	Publicidad, marketing con juegos, publicidad indirecta (publicidad por emplazamientos)	Uso de productos	Información de productos	Estudios de opinión	Machinima (corto de animación que usa un videojuego)	Estudios de opinión
Educación	Informar sobre enfermedades y riesgos sanitarios	Juegos sobre temática social	Formación de profesorado/ Entrenamiento de competencias específicas	Aprendizaje	Ciencias de la computación y Reclutamiento	Aprendizaje P2P. Constructivismo	Formación a distancia
Empresas	Información a empleados del sistema sanitario y Bienestar para los empleados	Educación y Concienciación del cliente	Formación de empleados	Formación continua y Cuantificación profesional	Publicidad/ Visualización	Planificación estratégica	Mando y Control
Industria	Prevención de riesgos laborales	Ventas y Contratación	Formación de empleados	Formación profesional	Procesos de optimización mediante simulación	Diseño Nano/Biotech	Mando y control

hasta campañas de concienciación. Pero sin duda, el dominio de aplicación que más destaca en el organismo de la salud, en cuanto a artículos publicados, es la ciberterapia y *exergaming*, también conocido como videojuegos para el ejercicio.

3.2. EXERGAMES

Como se ha comentado en la sección anterior, el dominio de aplicación que más destaca para el sector de la salud son los videojuegos para el ejercicio o por su término en inglés *exergames*. Lo que se pretende conseguir con estos *exergames* es incentivar la movilidad de ciertas partes del cuerpo o del cuerpo entero mediante el uso de juegos con ambientes interactivos para poder generar en la persona la motivación o estimulación suficiente para que pueda realizar dichos ejercicios. De esta forma, los *exergames* tienen como objetivo guiar al paciente para que realice de forma correcta los ejercicios planificados, y mejorar así también el número total de pacientes que ejecutan de forma adecuada sus planes de recuperación.

Con el avance de la tecnología han ido apareciendo distintas maneras de interactuar con estos entornos virtuales para la realización de los ejercicios. Estos modos de interacción

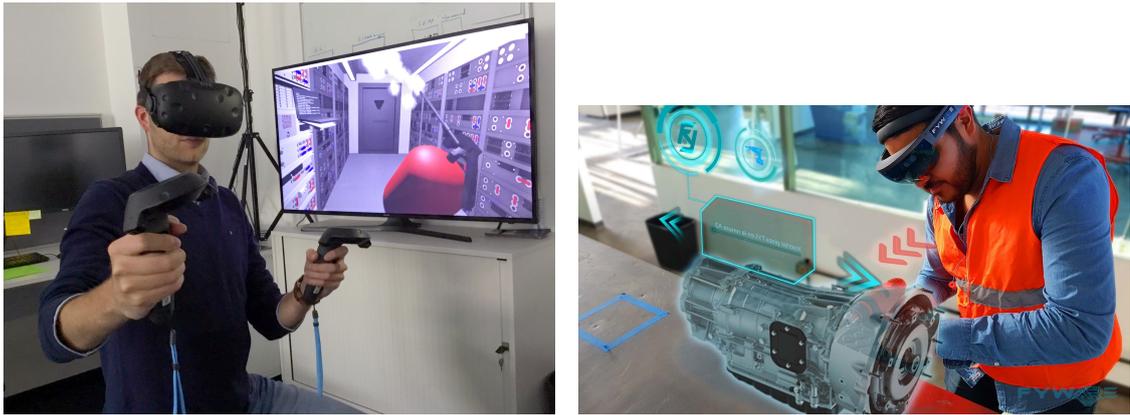
diferentes presentan beneficios pero también desventajas. Lo que se pretende realizar a continuación es una clasificación entre los diferentes modos de interacción en los entornos virtuales de rehabilitación.

Según el estudio realizado por *dos Santos, L. F.* [37], los modos de visualización en los entornos de rehabilitación virtuales se pueden dividir en seis grupos. El primero de los grupos que se expone es *avatar feedback*, en el que los movimientos que realiza la persona están representados por un avatar, ya sea de cuerpo completo o solo de partes del cuerpo. Dentro de este grupo se pueden distinguir tres subgrupos dependiendo de como se esté representando este avatar del cuerpo, pudiendo ser: a) realista, si la representación visual del cuerpo esta cercana a un cuerpo real; b) cuerpo reflejado, si el cuerpo es realista pero los movimientos se observan como si estuvieran reflejados en un espejo; y c) cuerpo simple, representación no real con algunos aspectos que lo asemejan a un cuerpo.

El siguiente grupo que se puede observar es *Augmented Reality feedback* donde la imagen de un usuario real es virtualizado en un entorno virtual [46]. El *Abstract feedback* donde los movimientos de la persona son representados como un esquema mediante gráficos por computador. Dentro de este grupo hay dos subgrupos dependiendo de las dimensiones del ejercicio y la representación pudiendo ser a) en dos dimensiones o b) en tres dimensiones. El siguiente tipo de *feedback* es el *Tracking feedback* en el que se realiza un ajuste continuo de los movimiento del cuerpo mediante una entrada visual externa. Otro tipo de *feedback* dado es el *Performance Evaluation feedback* donde el *feedback* visual solo se muestra una vez que el paciente ha realizado un ejercicio, indicándole datos sobre la realización del mismo. Y, por último, *Combined feedback*, donde se realiza una combinación de los diferentes tipos de *feedback* expuestos anteriormente.

Con el avance de las tecnologías de Realidad Aumentada y Realidad Virtual se diseñan más *exergames* centrándose en un *feedback* en entorno virtual, ya sea en entornos inmersivos o no inmersivos. Los entornos inmersivos envuelven al usuario completamente en un entorno virtual sin ninguna referencia al mundo real. Al contrario que los no inmersivos, en los que los usuarios siguen teniendo como referencia el mundo real, apareciendo de forma superpuesta los elementos virtuales.

En al Figura 3.1 puede observarse una comparación entre la realidad aumentada y la realidad virtual. En la Figura 3.1(a) se puede observar un uso recreativo de la realidad virtual. En este caso, el usuario está completamente inmerso en un mundo sintético, sin ninguna referencia con el mundo real. Sin embargo, en la Figura 3.1(b), donde se observa un uso industrial de la realidad aumentada, se puede observar como el usuario sigue teniendo contacto con el mundo real pero con elementos virtuales superpuestos, es decir, se esta «aumentando» la realidad.



(a) Persona usando VR, fotografía por ESA, CC BY- (b) Realidad aumentada, by Carlos Fy / CC BY-SA SA 3.0-igo, <https://commons.wikimedia.org/w/index.php?curid=62840930>, <https://creativecommons.org/licenses/by-sa/4.0/>

Figura 3.1: Comparación entre realidad virtual y realidad aumentada

3.2.1. Antecedentes

Esta sección ofrece una visión general sobre trabajos previos enfocados en el desarrollo de sistemas de ejercicios para la rehabilitación física o cognitiva a través de entornos de RA y RV, indicando el tipo de *feedback* que recibe el paciente, y el tipo de sistema de RA que usa.

MirrARbilitation

Este artículo [9] muestra un sistema de rehabilitación de RA con reconocimiento de gestos basado en un *tracking* corporal sin marcas. El *feedback* que recibe el usuario es su propio reflejo en un monitor con elementos superpuestos, por tanto, un *feedback* solo visual. Este sistema usa un *feedback* basado en avatar. El *tracking* del cuerpo del paciente se realiza utilizando el dispositivo *kinect* de *Microsoft*, de esta forma no será necesario añadir sensores en el paciente. El sistema guía y motiva al paciente durante la ejecución del ejercicio, donde los ejercicios son representados como un juego que consiste en tomar una pelota virtual y llevarla a la canasta. Los ejercicios son configurables permitiendo al terapeuta establecer los parámetros que debe seguir este *exergame*. El sistema proporciona una puntuación y da instrucciones para evitar la ejecución incorrecta de los movimientos.

Fruit Ninja

En el artículo [28] se presenta una versión del juego para móvil *fruit ninja* para realidad aumentada y PC para pacientes afectados por un accidente cerebro-vascular. En la versión de ordenador el paciente interactúa con el sistema a través del ratón y ve los resultados en pantalla, a diferencia de la versión de RA que está basada en proyecciones sobre una mesa, además el paciente también recibe un *feedback* sonoro. El sistema proporciona una puntuación de juego. Como resultado, se observó que los pacientes que jugaron a la versión de RA obtuvieron un 21 % más de puntuación que los que jugaron a la versión de PC, además,

fueron un 19 % mas rápidos en la ejecución de los ejercicios, que los que experimentaron con la versión de PC.

SleeveAR

En el artículo [42] se presenta un sistema que integra *feedback* multimodal para guiar al paciente durante la realización de los ejercicios de rehabilitación de miembros superiores. El sistema realiza unas proyecciones en el suelo que sirven de guía para el paciente para poder seguir el ejercicio de forma correcta, además informa en tiempo real si lo está haciendo bien o mal. *SleeveAR* esta diseñado para su uso en el hogar, pero actualmente, debido a su complejidad y su alto coste, *SleeveAR* está siendo usado en gimnasios de rehabilitación.

RehaBio

RehaBio [3] es un sistema de RA para la rehabilitación en casa usado para miembros superiores de pacientes que sufren hemiparesia debido a un accidente cerebrovascular, lesión cerebral traumática o lesión de la médula espinal. Los pacientes pueden realizar cuatro juegos: *ping pong rehab*, *ballon collection rehab*, *transfer object rehab* y *feeding animal rehab*. El *feedback* que reciben los pacientes es a través de un monitor donde se muestra el ejercicio y a través de los altavoces. El sistema también incluye la evaluación de los ejercicios que ha realizado el usuario.

Projection-Based AR

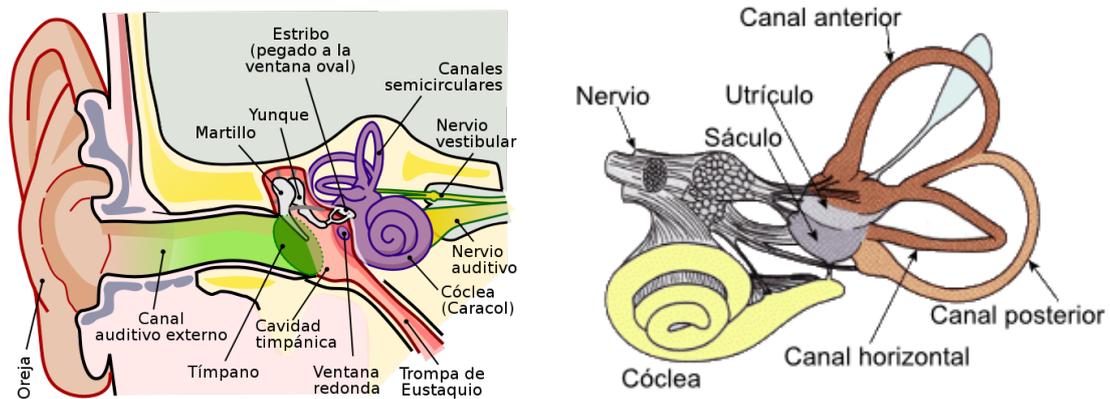
En el artículo [39] se expone un sistema de RA basado en proyecciones para la rehabilitación de la marcha. Este sistema pretende dotar a los pacientes de un mejor *feedback* visual del cuerpo y la cinemática del movimiento del paciente en los ejercicios de rehabilitación. Además, se muestra un experimento comparando sistemas de rehabilitación basados en monitor con el sistema que proponen obteniendo como resultado que los pacientes que hacían uso del sistema de RA presentaban una mejor coordinación pie-ojo.

3.2.2. ¿Realidad virtual o realidad aumentada?

Como se ha expuesto anteriormente, en los últimos años han habido diferentes propuestas de juegos serios para la rehabilitación que exploran diferentes entornos en los que el *feedback* que recibe el usuario era a través de entornos de RA o de RV. Se ha demostrado que utilizar estas tecnologías para la rehabilitación, tanto física como cognitiva, ayuda a que el paciente realice mejoras cognitivas mas rápidas y de forma más efectiva [36]. También para rehabilitaciones físicas se ha podido demostrar que el uso de tecnologías de RA y RV tienen un efecto positivo en el paciente respecto a las rehabilitaciones tradicionales [47]. Ahora, llegados a este punto cabe preguntarse cuál de los dos entornos es mejor para la

rehabilitación del paciente.

La RV ya ha sido explorada para contener juegos serios enfocados a la rehabilitación del paciente, ya sea rehabilitación física como rehabilitación cognitiva. Pero, sumergir al paciente en un mundo completamente sintético sin ninguna referencia con el mundo real puede conllevar efectos no deseados en los pacientes que están realizando estos ejercicios de rehabilitación. Los efectos negativos del uso de la inmersión en un mundo virtual son conocidos por su término inglés «*cybersickness*». El término *cybersickness* viene de la palabra anglosajona *cyber* (cibernético) y *sickness* (enfermedad), pero *cybersickness* no es una enfermedad, sino una respuesta fisiológica a un estímulo inusual [18]. El *cybersickness* incluye diferentes síntomas: mareos, náuseas, dolores de cabeza, malestar general, fatiga visual, aumento de la salivación, vómitos, somnolencia y otros menos comunes [17]. Estos síntomas aparecen durante o después de la exposición prolongada a un ambiente de RV.



(a) Diagrama del oído humano (cortesía de Chittka L, (b) Diagrama del sistema vestibular (cortesía de Thomas Haslwanter)

Figura 3.2: Sistema vestibular humano

Los síntomas del *cybersickness* se pueden dividir en dos grupos o categorías: los problemas oculares, causados mayormente por la fatiga visual consecuencia de tener los monitores muy cercanos a los ojos durante largos periodos de tiempo. Y, las náuseas, mareos y desorientación. Hay varias teorías que intentan explicar por qué se producen los síntomas de este segundo grupo, pero la teoría más aceptada en el campo de la realidad virtual es la *la teoría del desajuste sensorial* [5]. Esta teoría expone que los síntomas del segundo grupo del *cybersickness* tienen que ver con un conflicto entre el sistema vestibular del oído interno y los demás sentidos. En la Figura 3.2 se puede observar el oído humano 3.2(a) y el sistema vestibular 3.2(b). El sistema vestibular está localizado en el oído interno y se encarga de tres funciones principales: el movimiento ocular, la información vestibular ayuda a estabilizar los ojos durante los movimientos de la cabeza; la postura, la información recibida del sistema vestibular provoca ajustes en la actividad muscular y en la postura; y el balance, el sistema vestibular juega un papel muy importante en la sensación de movimiento y orientación en el espacio.

Durante la inmersión en un entorno virtual, puede producirse un conflicto entre la información percibida por el sistema vestibular y los otros sentidos como puede ser el de la vista. La *teoría del desajuste sensorial* sugiere que los síntomas conocidos por *cybersickness* surgen cuando la información enviada al cerebro a través de los canales semicirculares (Figura 3.2(b)) sobre la aceleración angular de la cabeza no guarda relación con la información enviada al cerebro de los demás sentidos.

Tal como se expone en [8], los problemas del *cybersickness* pueden ser solventados usando RA en vez de los entornos inmersivos de RV. La RA, al tener a los pacientes en contacto con el mundo real, permite que tengan interacciones más naturales e intuitivas proporcionando una experiencia más realista. Además, con la combinación de diferentes tipos de *feedback*, como se han comentado anteriormente, se consigue que la RA sea la mejor opción en el contexto de los *exergames* para la rehabilitación física. Igualmente, en el estudio [40] se trata de dar una respuesta a que entorno (RA o RV) ofrece un tratamiento más eficiente para la rehabilitación de miembros superiores. Para ello se diseñó un *exergame* parecido al juego *fruit ninja* (juego que ya ha sido usado para rehabilitación como se ha comentado en el apartado de antecedentes) con un grupo de personas sanas. Los resultados muestran que en el entorno de RA se obtuvieron mejores resultados que en el de VR en base a los siguientes criterios: los tiempos de respuesta en RA fueron menores que los de VR, lo que significa que los participantes que jugaron al juego en RA tuvieron mayor velocidad y mayor precisión que los que lo realizaron en VR. Dichos resultados son relacionados con la coordinación entre manos y ojos, es decir, mientras que los usuarios que utilizaban RA solo miraban sus manos, los que usaron VR tenían que mirar tanto la pantalla como sus manos. Por tanto, se puede concluir que los entornos basados en RA ofrecen ventajas significativas en el contexto de los *exergames*, frente a los entornos basados en RV.

El análisis anterior influye en la decisión de optar por métodos basados en RA en este proyecto. A partir de aquí, el estudio sobre diferentes alternativas para la definición de ejercicios de rehabilitación es fundamental para la consecución de objetivos. En el estudio [8] se muestran unas pautas que deberían seguir los *exergames* diseñados para RA para que sean seguros para los pacientes.

- Construir el juego desde una perspectiva en primera persona en un dispositivo de RA que mantenga al paciente en contacto con el mundo real. De esta forma se busca conseguir reducir lo máximo posible los efectos del *cybersickness*.
- Los controles no deben obstaculizar o interferir con los movimientos naturales del paciente. La forma de interactuar que tendrá el paciente con el sistema de RA debe ser cómoda, haciendo controles que no interfieran con sus movimientos lo máximo posible.
- Asegurarse de que las características del juego sean relevantes y estén alienadas con las acciones que deben ser evaluadas. Los juegos deben de tener concordancia con la

actividad que se pretende realizar.

- Evitar que la mano atravesase objetos virtuales. Esto es sobretodo relevante en juegos en los que el paciente tenga que interactuar con objetos virtuales a través de sus manos.
- Posibilidad de jugar sin necesidad de practica previa. El juego debe ser intuitivo para el paciente, que debería poder jugar sin necesidad de una explicación detallada o práctica previa.
- Los juegos se deben adaptar a la condición física o mental del paciente. Los juegos en los que el paciente debe alcanzar un objetivo pueden desmotivar al paciente en caso de que estos juegos sean demasiado difíciles para sus condiciones.
- Utilizar recompensas para aumentar el deseo del paciente para completar las tareas. Para ello se recomienda el uso de técnicas de gamificación que premien al paciente cuando esté realizando el ejercicio y aumentar así su motivación.

3.3. GAMIFICACIÓN

Como se ha comentado anteriormente, uno de los principales problemas de la rehabilitación física de los pacientes es la falta de motivación, la cual puede llevar al abandono de la terapia o a una recuperación más lenta [20]. El uso de juego serios para la rehabilitación también tiene como objetivo aumentar la motivación de los pacientes para evitar el abandono de los mismos y acelerar los plazos de recuperación. De ahí nace el concepto de «gamificación»

La primera aproximación al término «Gamificación» apareció en 2008 en un *blog* publicado por *Brett Terill* [44]. En él se describía el término como: “tomar las mecánicas del juego y aplicarlas a otras propiedades de la web para aumentar la interacción”. Desde esa primera aproximación al término han surgido otras formas de definir la gamificación, una de las más comúnmente aceptadas es: “el uso de elementos de diseño de juegos en contextos ajenos a los juegos” [11]. Las empresas han aceptado el concepto de gamificación como un medio para aumentar la retención de los usuarios.

Los juegos serios, como se ha comentado al inicio del capítulo, son juegos especialmente diseñados para provocar un cambio en el jugador, ya sea educativo, físico o cognitivo. Haciendo uso de técnicas de gamificación se pretende que estos juego serios sean más atractivos a los usuarios y desencadenen una motivación que los lleve a no abandonarlos.

En el estudio [6] se plantea la implementación de un juego serio para la rehabilitación en busca de aumentar la motivación y reducir el abandono de los ejercicios por parte de los pacientes afectados por una enfermedad neurológica. Los resultados fueron positivos, detectando mejoras en los pacientes y registrando un aumento de la motivación haciendo que ningún paciente abandonase el tratamiento y quisiera seguir el proceso de rehabilitación de este modo.

Otro estudio [12] pretende diseñar un juego serio de RV para mejorar la motivación de los pacientes que realizan la rehabilitación de miembros superiores. Se dividió a los pacientes en dos grupos, uno de control que realizaban rehabilitación tradicional y otro experimental que usaba el sistema de RV. Los resultados concluyeron que los pacientes que utilizaron el sistema de RV incrementaron su motivación comparado con el grupo de control.

Para reforzar más la motivación de los pacientes al hacer uso de los *exergames* se deberá hacer uso de la técnica de *gamificación* o *ludificación*.

3.4. ESTUDIO Y ELECCIÓN DE ENTORNOS DE RA Y DISPOSITIVOS HARDWARE

En esta sección se va a analizar y elegir el entorno y los dispositivos necesarios para el desarrollo del proyecto, siguiendo los requisitos expuestos con anterioridad. En primer lugar, se va a utilizar un entorno de RA para este proyecto al presentar menos tendencia a *cybersickness* debido a que mantiene al paciente en contacto con el mundo real. Además, se van a buscar técnicas de RA más avanzadas que las vistas en los antecedentes, debido a que existe un avance en las tecnologías de RA que permiten mayor interacción con el usuario y un mejor *feedback*.

En los sistemas comentados anteriormente se utilizaban varios tipos de *feedback* visual para representar la RA. En el caso de *MirrARbilitation* [9] la retroalimentación visual se realizaba a través de una pantalla. En este tipo de *feedback* el campo de visión es limitado, ajustándose solamente al tamaño del monitor donde se visualizan las imágenes. Otro de los problemas de este tipo de retroalimentación es la alineación espacial del usuario con respecto a la pantalla, además de la distancia entre la pantalla y el observador. Debido a estos problemas, este tipo de sistema de RA queda descartado para este proyecto.

Otro de los modos de visualización comúnmente usados es el basado en proyecciones, como es el caso de los sistemas *SleeveAR* [42] y el juego basado en *fruit ninja* [28]. Estos sistemas hacen uso de proyecciones frontales para mostrar imágenes directamente sobre superficies físicas. Estos sistemas mejoran a los basados en monitor, aumentando el campo de visión hasta un ilimitado teórico, una resolución escalable y una acomodación ocular mas sencilla. No obstante, este modo de visualización de RA también cuenta con problemas, tales como la proyección de sombras sobre objetos físicos interactivables por el usuario, restricciones del área de visualización (tamaño, forma y color de la superficie sobre la que se proyecta), y los proyectores convencionales solo se enfocan en un solo plano focal ubicado a una distancia constante, por lo que la proyección de imágenes en superficies no planas causará un desenfoque. Por estas desventajas también se descarta este modo de visualización para el presente proyecto.

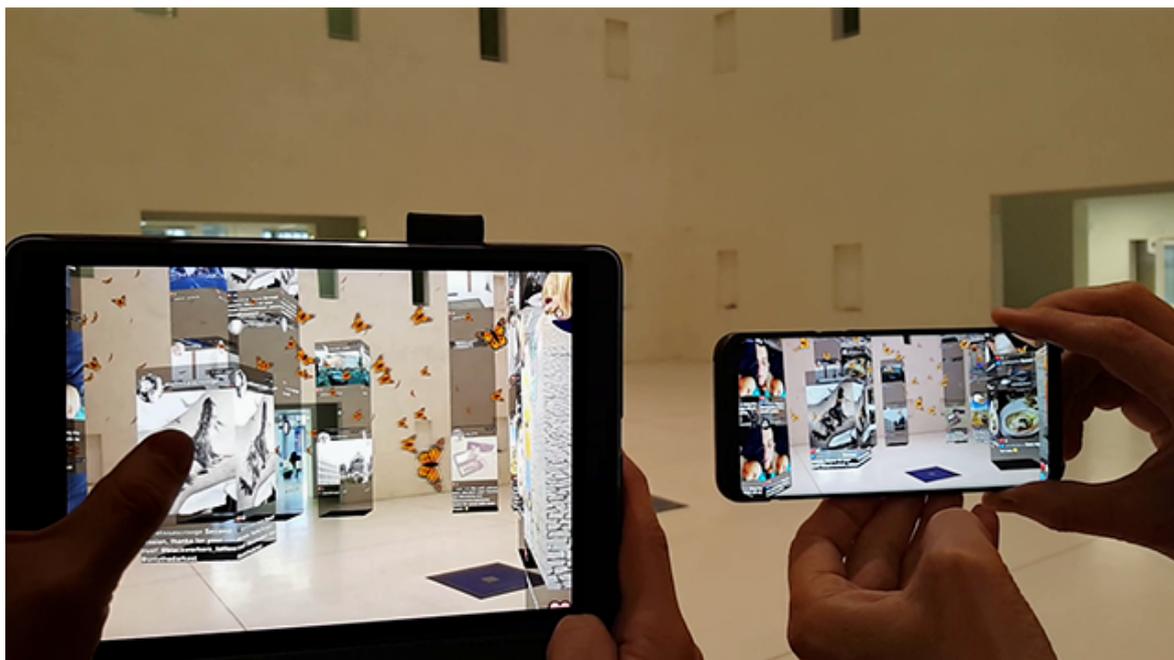


Figura 3.3: Ejemplo de realidad aumentada en el móvil con la aplicación *10.000 moving cities* (Marc Lee / CC 4.0)

Otro medio de visualización de RA empleado en la actualidad es el teléfono móvil (ver Figura 3.3). Esta figura muestra una aplicación de RA donde se pueden observar ciudades imaginarias que son destruidas y construidas por los usuarios, donde todos los usuarios ven lo mismo desde diferentes perspectivas.

Aunque el teléfono móvil es apto para ciertos contextos su uso no es el más adecuado para la rehabilitación. Esto es debido a que el usuario tendría que estar sujetando el móvil para poder recibir el *feedback* de la aplicación. Además, si se opta por usar un casco para colocar el móvil y transformarlo en un visor de realidad virtual contaría con los problemas de *cybersickness* que se han comentado anteriormente.

Por esta misma razón, los cascos de realidad virtual (HMD) suelen ser evitados para aplicaciones de rehabilitación debido a los efectos del *cybersickness*. Además, estas tecnologías de HMDs presentan también limitaciones tecnológicas, tales como: un pequeño campo de visión, el peso del dispositivo y la poca luminancia de los monitores.

Se espera que en los próximos años se lleven a cabo desarrollos de cascos de RA. más simplificados y mejorando sus funcionalidades. Actualmente nos encontramos con algunos HMDs de RA que cuentan con grandes ventajas que podrán ser usadas para el sistema que se plantea, como pueden ser las *Hololens* de *Microsoft* [25]. Las *Hololens* proporciona interacciones multimodales para una experiencia de RA más intuitiva y agradable.

3.4.1. Hologlens

Un aspecto fundamental por el que se ha decidido a usar las *Hologlens* (ver figura 3.4 en lugar de otro dispositivo de RA es por su independencia y capacidad de cómputo propia. Es decir, no necesitan estar conectadas a ningún otro dispositivo (PC o teléfono móvil inteligente) para funcionar, a diferencia de otros dispositivos de RA como, por ejemplo, las *Google glasses*. Además, las *Hologlens* son binoculares lo que se traduce en la posibilidad de «dibujar» objetos en tres dimensiones, a diferencia de las *Google Glasses* que, al ser monoculares, solo pueden mostrar objetos planos (bidimensionales).

Microsoft Hologlens es el primer dispositivo holográfico sin ataduras del mundo que combina dispositivos ópticos de última generación para ofrecer hologramas 3D anclados al mundo real. Posee dos lentes holográficas transparentes donde se proyectarán los hologramas, al ser las lentes transparentes se verá el mundo real.



Figura 3.4: Dispositivo *Microsoft Hologlens* By Ramadhanakbr - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=52546267>

Las *HoloLens* poseen una unidad de medición inercial (IMU) que contiene un acelerómetro, un giroscopio y un magnetómetro. Posee también cuatro sensores para «comprender» el entorno, una cámara de profundidad además de una cámara RGB de 2.4 megapíxeles. Cuenta con un procesador Intel de 32 bits y una unidad de procesamiento gráfico. Además, las *Hologlens* cuentan con un coprocesador diseñado por *Microsoft* exclusivamente para este dispositivo denominado *Microsoft Holographic Processing Unit* o HPU. El SoC y el coprocesador HPU cuentan con 1 GB LPDDR3 de memoria RAM y 8MB de SRAM compartida. Esta HPU usa 28 DSPs para procesar e integrar la información obtenida por los sensores con una gran rapidez.

Las *HoloLens* son un único dispositivo, y todo el esquema de control ha sido diseñado sobre esta cualidad. En vez de un teclado estándar, un ratón o una pantalla táctil, las *HoloLens*

utilizan tres formas primarias como mecanismos de entrada e interacción:

- **Mirada (*Gaze*).** Esta forma de interactuar se realiza a través de un cursor que se mueve con el movimiento de la cabeza. El cursor se proyecta y permite seleccionar aquello sobre lo que se fija el foco de atención. Es el equivalente al ratón en los ordenadores personales, en vez de mover el ratón, se mueve la cabeza. Un rayo es proyectado por el cursor permitiendo la interacción con los objetos virtuales al combinar con los gestos o la voz.
- **Gestos (*Gesture*).** A través de las cámaras y los sensores comentados anteriormente, las *HoloLens* están siempre analizando los gestos tales como *airtap* (juntar dedo pulgar y índice como si fuera una pinza) para simbolizar el *click*.
- **Voz (*Voiced*).** Mediante el uso del asistente personal *Cortana*, parte de *Windows 10*, y el sistema de reconocimiento de voz de *windows*.

Estas tres formas de interactuar con el entorno de RA se le conoce como *GGV*. Las *HoloLens* han sido diseñadas en base a estas tres formas de interacción. Con ellas se ofrece al paciente una forma adecuada de interacción sin tener que contar con teclados, mandos [31] o cualquier otro dispositivo que dificulte la realización de ejercicios de rehabilitación.

Desde que las *HoloLens* fueron lanzadas, aparecieron muchos juegos serios que hacían uso de esta tecnología en campos como la educación (ver figura 3.5, visualización de datos [21], turismo [10], medicina [35], arquitectura o ingeniería [49]). También esta empezándose a usar para tratar a pacientes con enfermedades degenerativas tales como la demencia o el Alzheimer [2]. Por estas razones se ha decidido usar como dispositivo para este sistema de rehabilitación física las gafas de realidad mixta de *Microsoft*. Aunque, este dispositivo cuenta con un problema, no ofrece una forma de obtener la posición de las articulaciones del paciente. Para solventar este problema se va a buscar otro dispositivo auxiliar para llevar a cabo esta tarea.

3.4.2. Dispositivo para el seguimiento de las extremidades

El único problema al que las *Hololens* no da solución es al sistema de *tracking* de las articulaciones del paciente. Como se ha comentado, las *Hololens* pueden reconocer gestos de las manos para poder interactuar con el entorno virtual, pero no puede obtener la posición tridimensional de ellas. Es importante obtener la posición tridimensional de las manos en el contexto de la rehabilitación puesto que, los ejercicios de rehabilitación se mueven en los tres ejes del espacio. Para solucionar este problema se va a buscar algún dispositivo capaz de obtener la posición de las articulaciones en el espacio. Es indispensable que pueda obtener la posición en el espacio tridimensional, ya que, si no, el sistema se limitaría solo a ejercicios bidimensionales, algo que sería una gran carencia en cuanto a ejercicios de rehabilitación. Una de las opciones para conseguir esta información es a través de la asignación de diferentes



Figura 3.5: Ejemplo de uso de las *HoloLens* para la educación. Imagen por Hoshinim / CC BY-SA (<https://creativecommons.org/licenses/by-sa/4.0>)

marcas en el cuerpo y, con una cámara y un *software* detectar estas marcas y así obtener la información de la posición de las articulaciones. Esta opción tiene la desventaja de tener que colocar marcas en el cuerpo del paciente para poder localizar sus extremidades, además que, con una *webcam* normal que toma imágenes bidimensionales, obtener la posición en la profundidad sería una tarea complicada y poco fiable, siendo también un problema si estas marcas se alejan mucho de la cámara o si son tapadas ya que el *tracking* se perdería. Por estas razones, el uso de marcas con cámaras queda descartado.



Figura 3.6: Dispositivo *kinect* para la consola *Xbox 360*

Otra opción es usar un dispositivo de *tracking* de cuerpo y uno de estos dispositivos con mas popularidad es el *kinect*. El dispositivo *kinect* fue desarrollado por *Microsoft* como un controlador de juego para la consola *Xbox 360*, se puede ver en la Figura 3.6 la primera versión del dispositivo y sus sensores. El dispositivo permitía interactuar con la consola sin necesidad de un controlador físico mediante una interfaz que reconocía los gestos del usuario y comandos por voz además de objetos e imágenes.

El dispositivo *kinect* en los videojuegos no tuvo el impacto esperado por *Microsoft*. Sin embargo, al ser un dispositivo de gran disponibilidad y bajo coste, muchos investigadores de diferentes especialidades tecnológicas empezaron a aprovechar esta tecnología en busca de nuevas formas de interactuar con las máquinas, además de su uso para el campo de la salud. *Microsoft* llamó a este evento *el efecto kinect* [50]. Este efecto aumentó después de que *Microsoft* lanzara el SDK para el dispositivo *kinect*. Este dispositivo funcionaba emitiendo señales infrarrojas a través del proyector de infrarrojos y calculando el tiempo que tardaba en volver y ser capturado por el sensor de infrarrojos (Figura 3.6). De esta forma se podía obtener la distancia que tenía un objeto con respecto de la cámara. Con estos datos se podía obtener la posición de las articulaciones de la persona [50].

Desde el lanzamiento del primer *kinect* en 2010 *Microsoft* ha lanzado nuevas versiones del dispositivo que lo mejoran. El *kinect 2.0* se lanzó en 2013 e incluía detección de hasta seis personas y 1080 píxeles de resolución. El último dispositivo *kinect* lanzado por *Microsoft* es el *azure kinect dk*. Las primeras versiones del *kinect* estaban pensadas para los videojuegos, esta nueva versión está pensada para su uso en el sector científico-tecnológico con sus grandes posibilidades para la logística, robótica, aplicaciones de salud, etc.

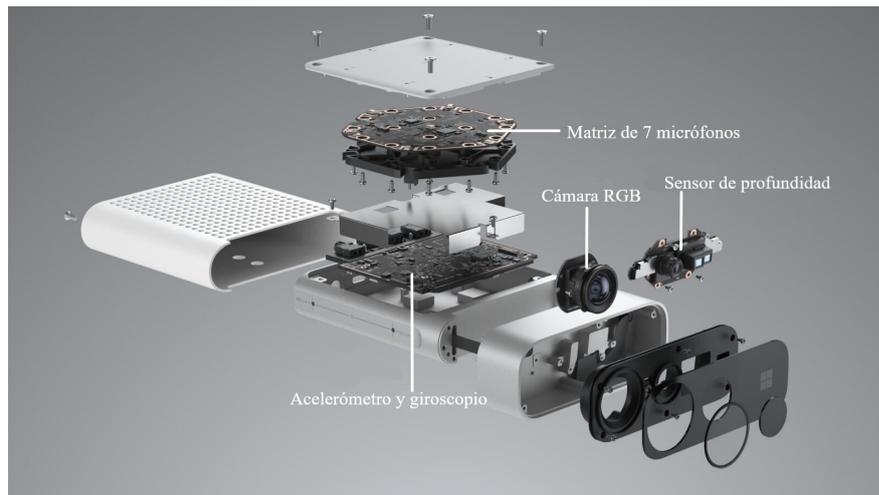


Figura 3.7: Azure kinect DK

El dispositivo *Azure Kinect* (ver Figura 3.7) presenta una cámara RGB de 21 megapíxeles que puede llegar a resoluciones de 3840x2160. La parte más importante para este trabajo es la cámara de profundidad. La cámara de profundidad de este dispositivo sigue el mismo principio que en las anteriores versiones del *kinect*. Esta cámara hace uso del principio del tiempo de vuelo (ToF) de onda continua modulada en amplitud. Es decir, se van a enviar señales en el espectro infrarrojo cercano y se va a registrar el tiempo que la señal está «en vuelo». A través de estas mediciones se podrá obtener el mapa de profundidad. La imagen infrarroja se transmitirá al PC Host, y a través del *software* y haciendo uso de una GPU si está disponible obtendrá el mapa de profundidad y con ello la posición de las articulaciones del paciente. Las articulaciones que se pueden obtener con el sistema de *body tracking* del

dispositivo *Azure kinect* se pueden observar en la Figura 3.8.

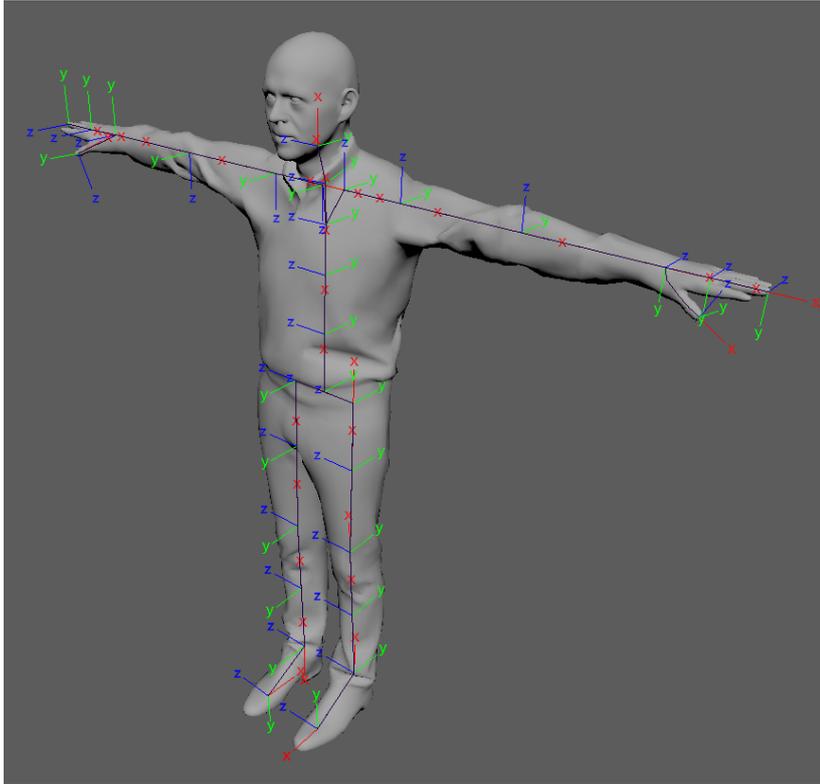


Figura 3.8: Articulaciones obtenidas por el dispositivo Azure Kinect. Documentación de Microsoft <https://docs.microsoft.com/en-us/azure/kinect-dk/body-joints>

METODOLOGÍA

Para un correcto desarrollo de un proyecto ingenieril es necesario seguir una metodología de trabajo que permita alcanzar los objetivos, de una manera eficiente y completa, que se propusieron al inicio del proyecto. En esta sección se va a tratar la metodología usada durante el de desarrollo del proyecto así como los recursos necesarios para la realización del mismo.

4.1. METODOLOGÍA DE TRABAJO

Este proyecto se puede dividir en cuatro partes. La aplicación de realidad aumentada para definir los ejercicios, la aplicación de realidad aumentada para realizar los ejercicios, la aplicación que va a permitir obtener la información del esqueleto del paciente y del *backend* donde se va a guardar la información de los ejercicios y sus resultados.

Debido al carácter de este proyecto, el cambio en los requisitos puede ser algo común al tratarse de un proyecto que necesita apoyo de rehabilitadores. Por lo que es adecuado poder adaptarse de forma rápida a esos cambios en los requisitos y, para ello, se ha decidido hacer uso de una metodología ágil. Usando una metodología ágil se va a permitir poder reaccionar de forma rápida a las valoraciones del proyecto durante su ciclo de desarrollo.

El *desarrollo iterativo e incremental*, también conocido como *iterativo y creciente*, es un tipo de metodología ágil para el cual se divide el trabajo en una serie de bloques temporales denominados iteraciones. Estas iteraciones durarán mientras dure el ciclo de vida del desarrollo. Se pueden ver las distintas fases de la metodología iterativa e incremental en la Figura 4.1. Lo primero que se puede observar es la primera fase de esta metodología, el *planteamiento inicial*. En esta fase se acuerdan los primeros requisitos del proyecto con el cliente (en este caso los tutores del trabajo), identificándose de esta manera el alcance del proyecto y los requisitos tanto funcionales como no funcionales. Una vez definido el planteamiento inicial se dan comienzo a las iteraciones. Las iteraciones constan de cinco fases: requisitos, análisis y diseño, implementación, pruebas y evaluación. Cada iteración

tiene como duración un pequeño periodo de tiempo (entre 2 y 4 semanas) y como resultado se tiene una versión del proyecto funcional que evoluciona de forma progresiva hasta el final del desarrollo. Cuando se termina una iteración, se analizan los resultados obtenidos y se comprueban que se han cumplido los objetivos pactados para esa iteración. Tras analizar los resultados de la iteración se empieza a planificar la siguiente iteración.

Para la realización de este proyecto se ha decidido seguir un desarrollo iterativo e incremental ya que se adapta perfectamente a los requisitos. Debido a que se puede realizar una división en cuatro partes, las primeras iteraciones han consistido en obtener una versión operativa de cada una de ellas, siendo las siguientes iteraciones, las encargadas de dotar de completa funcionalidad e integración a cada una de las partes. Antes de cada iteración se planifica una lista de tareas a realizar durante la iteración. Además de corregir los errores encontrados en la revisión de cada una de las iteraciones.

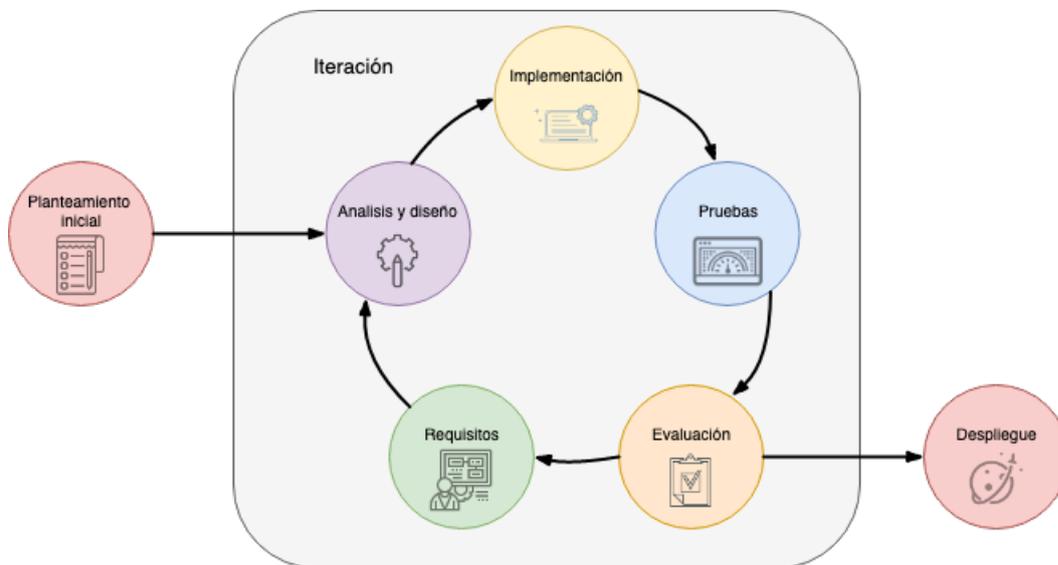


Figura 4.1: Esquema del desarrollo iterativo e incremental

4.2. RECURSOS

Para poder realizar el desarrollo del proyecto y cumplir con los objetivos pactados, serán necesarios recursos tanto *hardware* como *software*. A continuación, se van a enumerar los recursos utilizados junto con una breve explicación de los mismos.

4.2.1. Recursos hardware

Un aspecto fundamental de este proyecto es el *hardware* elegido para su desarrollo. La elección ya ha sido justificada en el capítulo 3. Además, también se van a mostrar los equipos utilizados en su desarrollo.

- *MacBook Pro*. Es el equipo usado para la mayor parte del desarrollo. Cuenta con 16GB de memoria RAM y una tarjeta gráfica *Radeon Pro 560* que cuenta con 4GB dedicados. Usado para el diseño e implementación de las aplicaciones de RA.
- *Asus R510V*. Es el equipo usado para la programación y las pruebas de la aplicación de *tracking* del esqueleto al contar con una tarjeta gráfica de *NVIDIA*, lo que va a permitir poder hacer uso de *CUDA* para un procesamiento mas rápido de la imagen del dispositivo de *tracking*.
- *Microsoft Hololens 1*¹: gafas inteligentes de realidad mixta desarrolladas y fabricadas por *Microsoft*.
- *Azure kinect DK*². Dispositivo desarrollado por *Microsoft* que cuenta con una serie de sensores y cámaras que van a permitir obtener la información de las articulaciones del paciente en un espacio tridimensional.

4.2.2. Recursos software

Sistemas operativos

- *Windows 10 Pro*. Este es el sistema operativo utilizado para todo el desarrollo de las aplicaciones de las *hololens* debido a que es el compatible y que nos va a permitir desarrollar y compilar sin problemas de compatibilidad. Además, también fue usado para el desarrollo de la aplicación del *kinect* encargada de obtener la información del esqueleto en el espacio.
- *Ubuntu 18*. Es el sistema operativo elegido sobre el que ha construido el *backend* debido a su facilidad de uso y versatilidad para usarlo en *docker*.

Herramientas de desarrollo

- *Unity*³: es un motor de videojuego que permite el desarrollo en un gran número de plataformas incluyéndose el dispositivo de realidad aumentada *Microsoft Hololens*, que es el dispositivo elegido para el desarrollo.
- *Visual Studio 2019*⁴. Es el editor usado para la programación de la aplicación de *Hololens* y *Kinect*. Es un IDE muy completo, recomendado por *Unity*, que permite una depuración simple y potente en los dispositivos.
- *Visual Studio Code*⁵. Es un editor de código fuente desarrollado por *Microsoft* utilizado para la programación del *backend*. Además, mediante un *plugin*, permitió el acceso mediante *SSH* a la máquina donde se encontraba el *backend*.

¹<https://docs.microsoft.com/es-es/hololens/hololens1-hardware>

²<https://azure.microsoft.com/es-es/services/kinect-dk/>

³<https://unity.com/es>

⁴<https://visualstudio.microsoft.com/es/vs/>

⁵<https://code.visualstudio.com>

- *NestJS*⁶: es un *framework* de *NodeJS* que permite construir de forma eficiente y escalable aplicaciones del lado de servidor de *NodeJS*. Integra *TypeScript* de forma nativa.
- *Docker*⁷: es un proyecto que permite la automatización y despliegue de aplicaciones dentro de contenedores de *software* de forma aislada. *Docker* se empleó para la construcción y el despliegue del *backend*.
- *PostgreSQL*⁸: es un sistema de gestión de base de datos relacional. La persistencia es muy importante en este proyecto, por ello, se hizo uso de una base de datos relacional a través de un sistema de gestión de base de datos *PostgreSQL*.
- *GIT*: es un *software* de control de versiones. Para, además de tener control de versiones poder tener un respaldo en la nube se empleó *Github* para el desarrollo.

Lenguajes de programación

- *C#*: es un lenguaje de programación desarrollado por *Microsoft* que sigue el paradigma de la programación orientada a objetos. Es el lenguaje de programación usado por *unity* para la elaboración de los *scripts* de comportamiento.
- *C++*: es un lenguaje de programación diseñado con la intención de extender el lenguaje de programación *C*, añadiendo mecanismos que permitieron la programación orientada a objetos. *C++* es un lenguaje multiparadigma que permite tanto programación estructurada como programación orientada a objetos. Fue el lenguaje usado en la programación del *kinect*.
- *TypeScript*: es un lenguaje de programación de código abierto que ha sido desarrollado y mantenido por *Microsoft*. Este lenguaje extiende la sintaxis de *JavaScript* añadiendo tipos estáticos y objetos basados en clases. Este lenguaje fue utilizado para la construcción del *backend*.
- *Bash*: es un lenguaje de comandos y *shell* de Unix. Su uso fue destinado a los *scripts* de arranque y mantenimiento del *backend*.
- *JSON*: es un formato de texto sencillo para el intercambio de datos que se utilizó para el intercambio de información entre la aplicación y el *backend*.

Kits de desarrollo software y bibliotecas

- *Azure kinect Sensor SDK*⁹: kit de desarrollo que contiene todas las bibliotecas necesarias y los *wrappers* para poder acceder y manejar los sensores y dispositivos del *Azure Kinect DK*.
- *Azure Kinect Body Tracking SDK*¹⁰: kit de desarrollo que permite construir una aplicación haciendo uso de la detección de cuerpos del dispositivo *Azure Kinect DK*. Este

⁶<https://nestjs.com>

⁷<https://www.docker.com>

⁸<https://www.postgresql.org>

⁹<https://docs.microsoft.com/en-us/azure/kinect-dk/sensor-sdk-download>

¹⁰<https://docs.microsoft.com/en-us/azure/kinect-dk/body-sdk-download>

SDK hace uso de *CUDA* para mejorar su rendimiento.

- *SDK Windows 10.0.18362.0* : kit necesario para la compilación y despliegue de la aplicación para las *Hololens*.
- *Mixed Reality Toolkit*¹¹: es un proyecto impulsado por *Microsoft* que proporciona las herramientas necesarias para acelerar el desarrollo de aplicaciones VR/AR en *Unity*.
- *Vuforia Engine*¹²: es un kit de desarrollo orientado a la realidad aumentada para *Unity* que nos va a permitir poder reconocer marcas, algo que será útil en este proyecto.

Documentación

- \LaTeX . Es un sistema de composición de texto de alta calidad especialmente usado para la elaboración de documentos de carácter científico-tecnológico. El editor usado para la escritura en \LaTeX ha sido *Overleaf*¹³, un *software online* que dota de un *IDE* completamente equipado para la escritura, compilación y visualización de \LaTeX .
- *Draw.io*¹⁴. Aplicación web y de escritorio que permite crear todo tipo de esquemas. Se ha usado para la creación de los esquemas de este documento.

¹¹<https://github.com/microsoft/MixedRealityToolkit-Unity>

¹²<https://developer.vuforia.com>

¹³<https://overleaf.com>

¹⁴<https://draw.io>

ARQUITECTURA

En este capítulo se va a realizar una descripción técnica del sistema *PhyRe UP!*, viendo cómo se ha desarrollado, los problemas que han surgido y cómo se han resuelto. En primer lugar, se va a mostrar una visión general del sistema y de las divisiones del mismo, para luego centrarse en cada una de las partes y en sus capas, analizando los problemas de cada una y las soluciones adoptadas.

5.1. VISIÓN GENERAL

PhyRe UP! es un sistema dividido a su vez en cuatro aplicaciones bien diferenciados como se puede observar en la Figura 5.1. La primera aplicación es la encargada de hacer el *tracking* de las articulaciones del paciente, información que luego es enviada a la aplicación de realidad aumentada que se esté usando. *PhyRe UP!* cuenta con dos aplicaciones de realidad aumentada, una para el terapeuta y otra para el paciente. Para la persistencia en la aplicación se ha decidido hacer uso de un *backend* que cuente con una *API Rest* y una base de datos.

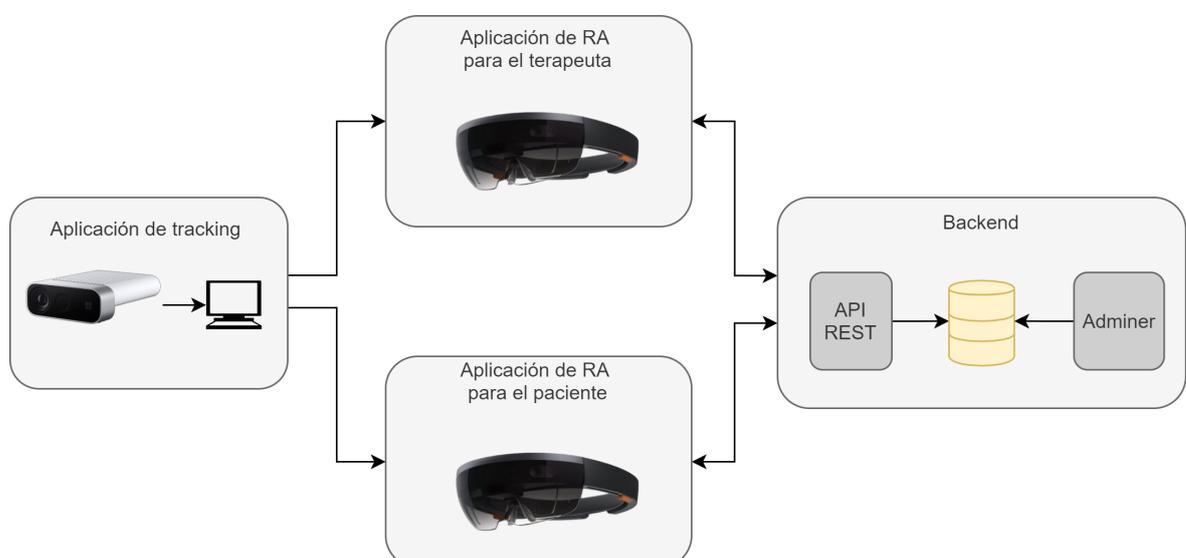


Figura 5.1: Esquema de los sistemas que componen *PhyRe UP!*

Las cuatro aplicaciones del sistema *PhyRe UP!* se muestran en la siguiente lista:

- *Aplicación para el tracking del cuerpo.* Se encarga de obtener la información del esqueleto. Esta información se envía a la aplicación de realidad aumentada para que pueda obtener la posición de cada articulación en el espacio.
- *Aplicación de RA para la definición de ejercicios.* La aplicación la usa el terapeuta para definir los ejercicios que van a realizar los pacientes, definir las rutinas y asignar estas rutinas a los pacientes.
- *Aplicación de RA para la realización de los ejercicios.* La utiliza el paciente para su rehabilitación. En esta aplicación, el paciente puede ver los ejercicios asignados y realizarlos. Tras la realización de cada ejercicio recibe un comentario de evaluación sobre los resultados.
- *Backend.* Se trata de la parte del sistema encargada de guardar los ejercicios, las rutinas y los resultados de la realización de los ejercicios. Para ello se ha desarrollado una *API REST* y una base de datos con las funciones necesarias para el funcionamiento del sistema *PhyRe UP!*.

5.2. APLICACIÓN DE TRACKING DEL CUERPO

Como se ha comentado en el Capítulo-3, para el desarrollo del sistema *PhyRe UP!* es necesario realizar el *tracking* de las articulaciones del paciente en el espacio 3D. Además, se busca que este *tracking* se realice sin necesidad de hacer uso de sensores o marcas que deban ser colocadas en el cuerpo del paciente, debido a que esto puede tener desencadenar dificultades en el paciente a causa de la sobrecarga que les pueda provocar. Para solucionar este problema, se ha decidido usar el dispositivo *Azure Kinect DK*, que, como se ha analizado en el Capítulo-3, es una de las mejores opciones actualmente para esta función.

5.2.1. Visión general

La aplicación está compuesta de cinco módulos bien diferenciados como bien puede verse en la Figura 5.2. El primer módulo es el encargado de manejar el dispositivo *Azure kinect*. El siguiente módulo se encargará de obtener las capturas o *frames* y enviárselas a la cola de procesamiento de la librería *Azure kinect body tracking*. Cuando se haya analizado la imagen y obtenido la información de las articulaciones se enviará el resultado a la cola de resultados. Una característica fundamental de esta aplicación es el rendimiento, para ello se ha estudiado si es necesario hacer uso de hilos que vayan leyendo estos datos puestos en la cola de resultados, realizar la transformación y enviarlo a la aplicación principal. Cabe destacar que es necesario obtener una aplicación eficiente en este proceso, ya que el *feedback* que va a recibir el usuario se verá muy afectado si no se realiza este procesamiento acorde al movimiento del paciente o terapeuta.

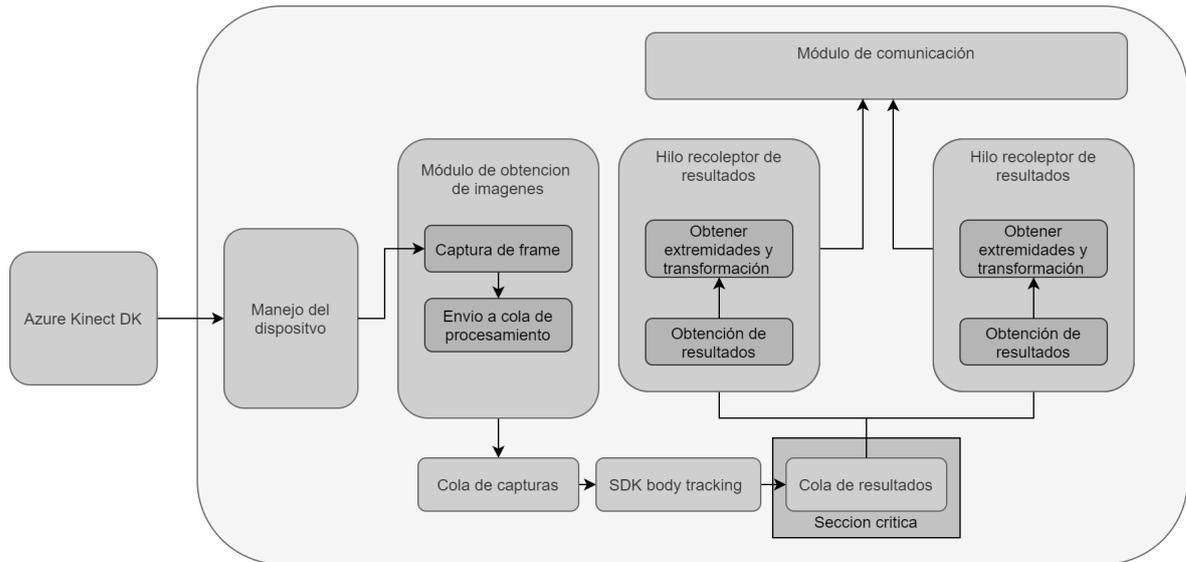


Figura 5.2: Esquema de la aplicación de *tracking* del esqueleto

5.2.2. Inicio y acceso al dispositivo

El dispositivo *Azure kinect* ofrece diferentes modos de funcionamiento con diferentes resoluciones de cámara. Para la función necesaria en este proyecto, que es el *tracking* del cuerpo, se hará uso de la cámara de profundidad. Como bien se ha explicado en el capítulo del estado del arte cuando se analizaban los diferentes dispositivos, el dispositivo *Azure kinect* cuenta con dos cámaras, la cámara RGB y la cámara de profundidad o *depth camera*. Para poder hacer uso de la función de *tracking* del esqueleto será necesario usar la cámara de profundidad. Esta cámara proyecta una iluminación modulada en el espectro IR cercano (NIR), que posteriormente registra el tiempo que tarda la luz en viajar de la cámara a la escena y viceversa, lo que permite obtener una imagen tridimensional del entorno.

La cámara toma las imágenes IR moduladas sin procesar y las envía al *PC Host* que será encargado de crear el mapa de profundidad de la imagen. Para ello, el kit de desarrollo de *Azure body tracking* hace uso de la GPU, si está disponible, para generar este mapa de profundidad de forma rápida, aunque también puede trabajar solo usando la CPU. Esto se ha contemplado en el diseño de la solución permitiendo tanto hacer uso de una GPU *Nvidia* integrada, en caso de que se disponga, o hacer sólo uso de la CPU, lo cual no es recomendable debido al bajo rendimiento.

Tabla 5.1: Diferentes modos de funcionamiento de la cámara de profundidad del *Azure kinect*

Modo	Resolución	FoI	FPS	Rango de operación	Tiempo de exposición
NFOV unbinned	640 x 576	75° x 65°	0, 5, 15, 30	0.5 - 3.86 m	12.8 ms
NFOV 2x2 binned	320 x 288	75° x 65°	0, 5, 15, 30	0.5 - 5.46 m	12.8 ms
WFOV 2x2 binned	515 x 512	120° x 120°	0, 5, 15, 30	0.25 - 2.88 m	12.8 ms
WFOV unbinned	1024 x 1024	120° x 120°	0, 5, 15	0.25 - 2.21 m	20.3 ms

La cámara de profundidad del dispositivo *Azure Kinect* soporta varios modos de funcionamiento como se puede ver en la Tabla 5.1. En ella se muestran las formas de funcionamiento, junto con la resolución de la imagen, los fotogramas por segundo, el rango de operación y el tiempo de exposición¹. Un aspecto clave en este proyecto es el tiempo de respuesta, ya que este tiempo de respuesta influye directamente, y de forma considerable, en el *feedback* que van a recibir los usuarios. Por ello hay que buscar la mejor configuración del dispositivo que permita tener los mejores tiempos de respuesta. Aunque no todos los modos mostrados en la Tabla 5.1 son soportados para el *tracking* del esqueleto. Los únicos modos en los que podrá funcionar el sistema son: *NFOV Unbinned*, *NFOV 2x2 Binned*, *WFOV Binned* y *WFOV Unbinned*. No obstante, se va a descartar el modo *WFOV Unbinned* al poder trabajar como máximo a 15 FPS, algo que reduciría de forma notable el rendimiento.

En el inicio del *software* se podrá elegir en qué modo va a operar la cámara de profundidad y si se va a poder usar una GPU de *Nvidia* para realizar el tratamiento de la imagen. Para ello se va a poder indicar los parámetros por línea de comandos, se ha decidido hacerlo de este modo para permitir una máxima capacidad de configuración. La configuración se define en un *struct*, que contendrá la información que se puede ver en el Listado 5.1. Los parámetros que se aceptan por argumentos son:

- *IP <direccion_ip>*. Parámetro que indica la dirección IP del dispositivo *Hololens* al que se le van a pasar los datos de la posición de las articulaciones.
- *CPU*. Parámetro opcional. Este parámetro indica que solo se va a hacer uso de la CPU para tratar la imagen. Solo debe usarse cuando el PC *Host* no cuenta con una tarjeta gráfica compatible con *CUDA* [29].
- *<Modo de la cámara de profundidad>*. Indica el modo en el que va a funcionar la cámara de profundidad. Estos modos pueden ser: *NFOV_UNBINNED*, *NFOV_BINNED* y *WFOV_BINNED*.

Listado 5.1: Estructura usada para almacenar los parámetros de configuración

```

1
2 struct InputSettings {
3     k4a_depth_mode_t DepthCameraMode = K4A_DEPTH_MODE_NFOV_2X2BINNED;
4     bool CpuOnlyMode = false;
5     std::string serverIP;
6 };

```

Para la apertura y manejo del dispositivo se requieren las funciones del SDK. Primero se abre el dispositivo y a partir de ahí se configura con los parámetros vistos anteriormente. Una vez iniciada la cámara se configura el *tracker* que es el encargado de transformar la imagen IR a el mapa de profundidad y obtener la información de las articulaciones.

¹Cantidad de tiempo que permanece abierto el obturador de la cámara dejando pasar la luz hasta el sensor

5.2.3. Obtención de la posición de las articulaciones

Lo más importante de esta parte del sistema es la obtención de los datos del esqueleto. Para el diseño de esta solución se ha optado por un procesado asíncrono de los *frames*. Se ha decidido hacer el procesado asíncrono debido a que, como se ha comentado anteriormente, el rendimiento es un aspecto fundamental. Un procesado síncrono toma un *frame* y luego lo procesa, una vez que lo ha procesado vuelve a tomar otro *frame*. Esto provocaría un gran retraso al tener que esperar a que se complete todo el procesado antes de volver a capturar otro *frame* y, como consecuencia, los usuarios recibirán un peor *feedback*. Al contrario que un procesado asíncrono en el que una parte del sistema se encarga solo de tomar las imágenes, mientras que otra parte las va procesando. Para realizar este procesamiento asíncrono, primero se captura la imagen y después se añade a una cola de procesamiento. Las imágenes de esta cola las obtendrá la librería *Azure Kinect Body Tracking* que se encargará de obtener el mapa de profundidad de la imagen y posteriormente identificar las extremidades. No es necesario ningún mecanismo de protección a la cola puesto que solo el hilo principal encolará los *frames*.

Para la siguiente parte del *software* se ha pensado en la forma más eficiente de obtener la información del esqueleto, realizar una pequeña transformación y posteriormente enviar los datos al dispositivo *HoloLens*, donde funciona la aplicación de RA. Para buscar la arquitectura que mejor rendimiento obtenga, se han realizado unas pruebas con diferentes configuraciones. Para ello se ha usado un pequeño *script* en *Python* que simula el dispositivo *HoloLens*, creando un servidor para recibir la información de la posición de las manos. Recibe cien *frames* y se obtiene el tiempo transcurrido desde la primera recepción a la última. Se ha repetido la prueba doce veces para cada combinación, se han eliminado los valores pico y se hace la media entre los diez valores restantes para poder obtener el tiempo promedio en obtener la información de cien *frames* y enviarlo al dispositivo destino.

Tabla 5.2: Tabla con resultados de comparación con diferentes arquitecturas

	0 hilos	1 hilo	2 hilos
NFOV Unbinned	6,66s	6,66s	6,52s
NFOV Binned	5,76s	4,19s	4,07s
WFOV Binned	10,04s	9,01s	8,14s

Tras analizar los resultados que se pueden observar en la Tabla 5.2, se concluye que el mejor rendimiento se obtiene cuando se hace uso de dos hilos para obtener la información de la cola de resultados. Por esa razón se ha decidido usar esta configuración de dos hilos. Como esta cola es un recurso compartido, se ha de realizar el acceso en exclusión mutua. Para ello se ha hecho uso de un semáforo *mutex* que se bloqueará cuando se vaya a acceder a la cola y se desbloqueará cuando se haya obtenido un resultado. Esto se puede ver en el Listado 5.2.

Listado 5.2: Hilo de obtención de resultados

```

1 while (s_isRunning) {
2     k4abt_frame_t bodyFrame = nullptr;
3     sem_queue.lock();
4     k4a_wait_result_t popFrameResult = k4abt_tracker_pop_result(tracker, ↵
5     ↵ &bodyFrame, 0);
6     sem_queue.unlock();
7     if (popFrameResult == K4A_WAIT_RESULT_SUCCEEDED) {
8         int num_bodies = k4abt_frame_get_num_bodies(bodyFrame);
9         if (num_bodies > 0) {
10            k4abt_skeleton_t skeleton;
11            k4abt_frame_get_body_skeleton(bodyFrame, 0,&skeleton);
12            udpClient.send_data(processResultAndSend(skeleton));
13        }
14    }
15 }

```

El resultado de la imagen tratada es un mapa de cuerpos, teniendo cada cuerpo un mapa con las diferentes articulaciones. En el caso de esta aplicación solo se obtiene la información de un cuerpo que aparece ante la cámara. Cada articulación cuenta con las coordenadas en el espacio, un cuaternión [7] con las rotaciones de esa articulación y el nivel de confianza. Hay cuatro niveles de confianza que van del menor nivel de confianza, en el que la articulación está fuera de rango y por tanto no se pueda obtener con fiabilidad su posición, a el mayor nivel de confianza el cual significa que la posición de la articulación es muy confiable. Las articulaciones que se obtienen se muestran en la Figura 5.3

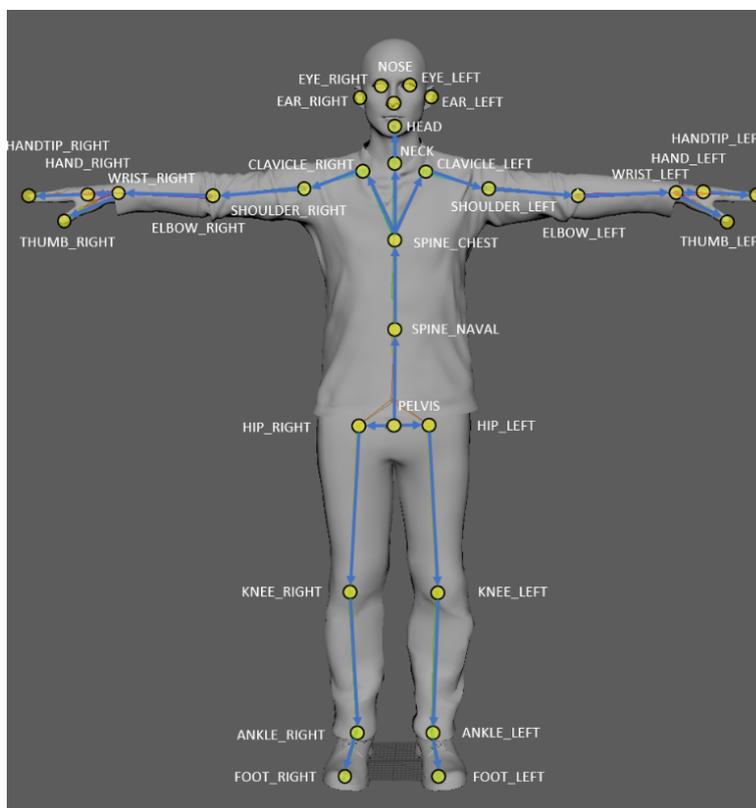


Figura 5.3: Articulaciones obtenidas por el dispositivo Azure Kinect. Documentación de Microsoft

Para la aplicación que se quiere presentar con este trabajo, se obtiene la información de las manos derecha e izquierda. La información de la posición tridimensional de las articulaciones que se obtiene desde el dispositivo *Azure kinect* se representa en milímetros, mientras que las distancias en la aplicación de las *Hololens* se representa en metros. Además, el eje de coordenadas que usa *Azure kinect* es inverso al de las *HoloLens* contando con que el dispositivo *Azure kinect* va a estar delante del paciente o terapeuta. Dadas estas circunstancias, se realizará un cambio de unidades antes de enviar esta información a las *Hololens* y una transformación del eje de coordenadas.

5.2.4. Comunicación con la aplicación de RA

La comunicación es un punto importante debido a que cualquier retraso perjudicaría de forma significativa la experiencia de los usuarios. Por lo que se pretende buscar la mejor configuración para enviar los datos, premiando la velocidad. Para la comunicación de los datos obtenidos por la aplicación de *tracking* del cuerpo y la aplicación de RA que usa el usuario en el dispositivo *Microsoft HoloLens*, se ha decidido usar *socket* y el protocolo UDP. Se ha optado por este protocolo al ser más rápido que TCP para aplicación que aquí se concibe [48], y también porque la comunicación es solo *one-way*. Una de las desventajas de usar el protocolo UDP es la pérdida de datagramas, aunque la pérdida de algún punto casual es poco probable y no es un hecho importante. Si este punto se reenviara en caso de perderse no se realizaría una mejora, es más, podría empeorar el *feedback* al volver la posición a un punto anterior que ya carece de importancia al estar en el pasado.

5.3. APLICACIÓN RA PARA EL TERAPEUTA

En el sistema *PhyRe UP!* se pretende que el terapeuta pueda, a través de un aplicación de RA en el dispositivo *HoloLens*, definir los ejercicios y las rutinas que los pacientes van a realizar.

En la Figura 5.8 se muestra una vista general de la arquitectura de la aplicación con una división en módulos. La división modular va a permitir que varios módulos de esta aplicación sean reutilizados en la aplicación del paciente, lo cual ha ayudado a acelerar el proceso de desarrollo. Estos módulos que son usados tanto en la aplicación del terapeuta como en la del paciente solo serán explicados una vez.

También puede distinguirse en la arquitectura los otros dos subsistemas, la aplicación encargada del *tracking* de las articulaciones del paciente y el *backend* encargado de dotar de persistencia al sistema.

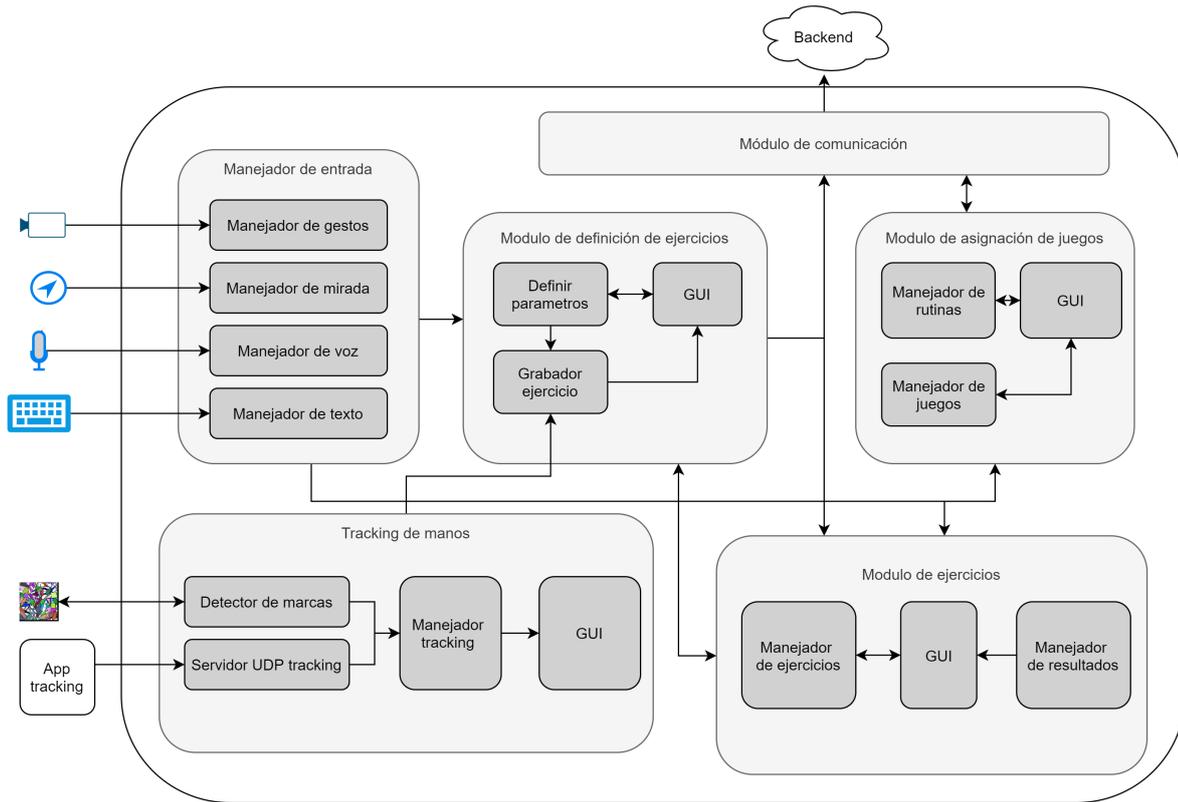


Figura 5.4: Vista general de la aplicación de RA para el terapeuta

5.3.1. Entradas

Las formas de interactuar con el entorno de RA se realizan usando diferentes modalidades que ofrece el dispositivo *Microsoft HoloLens*.

- Gestos. *PhyRe UP!* utiliza el gesto de «click» para poder interactuar con los botones que componen la interfaz de usuario. Pese a que el dispositivo *HoloLens* permite capturar más gestos para la interacción, no se han usado con el fin de hacer más sencillo la forma de interactuar con el sistema.
- Mirada. A través del giroscopio y acelerómetro, las *HoloLens* permiten trazar un rayo que apunte hacia donde el usuario está mirando. Esta forma de interacción será usada para la navegación por los distintos menús de la interfaz.
- Voz. A través del reconocimiento de voz de *Microsoft* se puede interactuar con el sistema usando comandos de voz.
- Texto. Para la entrada de texto se hace uso de un teclado virtual que aparece sobre la aplicación de RA. Para interactuar con este teclado se hace uso del gesto «click» y del rayo trazado por la mirada para seleccionar la tecla.

5.3.2. Tracking de manos

Esta es la parte del sistema encargada de recibir la información de la posición de las manos que son capturadas por la aplicación de *tracking* del cuerpo. Cuando el usuario inicia la aplicación de RA, se crea el eje de coordenadas en la posición actual del dispositivo *HoloLens* y se queda anclado a la habitación. Lo que significa que independientemente de los movimientos que realice el usuario, el origen de coordenadas estará fijo en el punto de la habitación donde se inició la aplicación. Los ejes de coordenadas que se anclarán al inicio se observan en la Figura 5.5(a), cabe destacar que el usuario estaría mirando hacia el eje «z» positivo, mientras que el *Azure Kinect* hacia el eje «z» negativo.

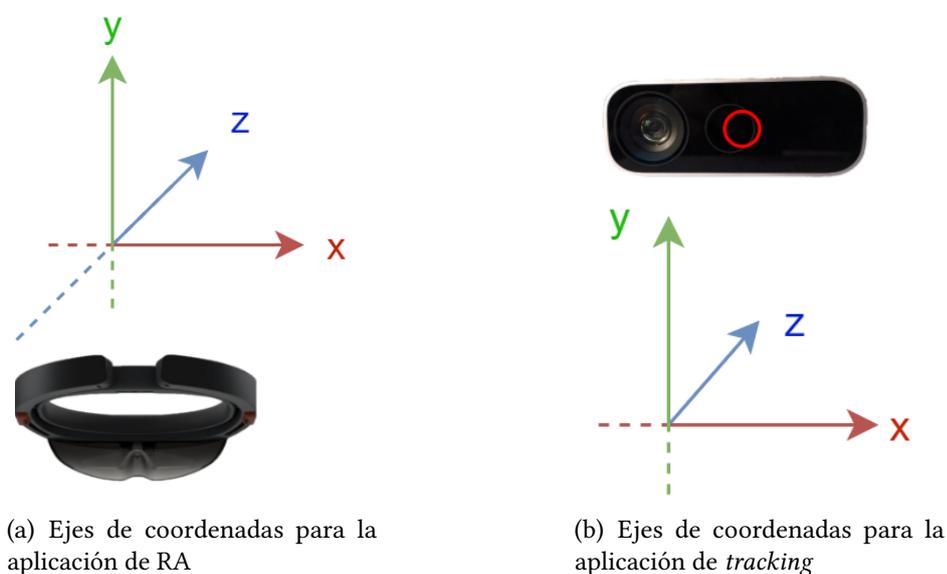


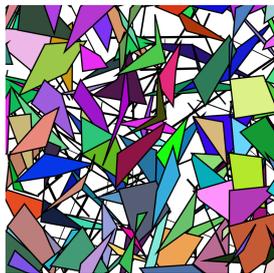
Figura 5.5: Ejes de coordenadas para la aplicación de realidad aumentada y para la de *tracking* del cuerpo

El problema surge con la información que llega al dispositivo de RA desde la aplicación de *tracking*. Esta información, que contiene un punto en el espacio donde está la articulación, tiene como origen de coordenadas el dispositivo de *tracking* del cuerpo. Para poder solucionar este problema se pretende anclar un sistema de coordenadas dentro del espacio virtual en la posición del sistema de *tracking*. Para realizar esta operación se necesita saber la posición del sistema de *tracking* del cuerpo en el espacio virtual. Esta tarea debe hacerse de forma que el usuario tenga la menor interacción posible, puesto que, una operación de calibrado compleja puede suponer un gran problema para usuarios del sistema.

Una de las posibles soluciones es hacer uso del sistema de mirada (*gaze*) que presentan las *HoloLens*. De esta forma solo sería necesario mirar al dispositivo para obtener la posición del dispositivo cuando el puntero «choque» con un obstáculo real, en este caso, el dispositivo de *tracking*. Esta solución parece sencilla, pero cuenta con importantes problemas. El primer problema que se presenta es que depende de que las *HoloLens* hayan podido realizar un

buen *mapeo espacial*². A través de la cámara de profundidad se realiza un mapeo de las superficies del mundo real, de esta forma se permite colocar objetos virtuales sobre objetos reales creando una realidad aumentada que se asemeja más a la realidad. El problema de hacer uso del sistema de mapeo espacial reside en que el tamaño del dispositivo de *tracking* es relativamente más pequeño que los objetos del entorno por lo que podría pasar desapercibido. El segundo problema reside en que no se puede obtener la rotación del dispositivo con respecto al origen de coordenadas del sistema virtual de las *HoloLens*. Esto es importante ya que, si no se contempla la posibilidad de que el sistema de coordenadas del dispositivo de *tracking* pueda estar rotado, el usuario siempre tendría que iniciar la aplicación «mirando» al dispositivo. Para que, de esta forma, cuando la aplicación de RA inicie, establezca el eje de coordenadas alineado al del dispositivo de *tracking*. Esto podría provocar que, si el usuario gira la cabeza mientras inicia la aplicación, los ejes de coordenadas no estuvieran alineados y la aplicación no funcione de forma correcta ya que no se podría obtener la rotación del dispositivo para poder establecer su sistema de coordenadas correctamente. Por esta razón, se ha decidido no hacer uso de esta solución.

Gracias a la capacidad de cómputo y sensores del dispositivo de RA *Microsoft HoloLens*, es posible usar técnicas de visión por computador para el reconocimiento de marcas que permitan obtener la posición y rotación del dispositivo de *tracking* en tiempo real. Para realizar esta tarea se va a hacer uso del kit de desarrollo *software Vuforia Engine* [43]. *Vuforia Engine* utiliza tecnología de visión por computador para reconocer y rastrear imágenes planas y objetos 3D en tiempo real. Por ello, se ha diseñado una marca (ver Figura 5.6), que será usada para obtener la posición y rotación del dispositivo de *tracking*.



(a) Marca para el dispositivo de *tracking*



(b) *Azure kinect* con la marca

Figura 5.6: Marca para obtener la posición y rotación del dispositivo de *tracking*

²<https://docs.microsoft.com/en-us/windows/mixed-reality/spatial-mapping>

En la Figura 5.7 se puede observar el funcionamiento. El dispositivo de RA, a través del SDK *Vuforia Engine*, obtiene la posición y la rotación del dispositivo de *tracking*. Una vez obtenida su posición se crea en esa posición un origen de coordenadas, aplicando también las rotaciones detectadas en la marca para posicionar de forma correcta el nuevo eje de coordenadas. La información de las articulaciones se tomará desde el sistema de coordenadas que se ha creado, teniendo así la posición correcta de la articulación *trackeada*.

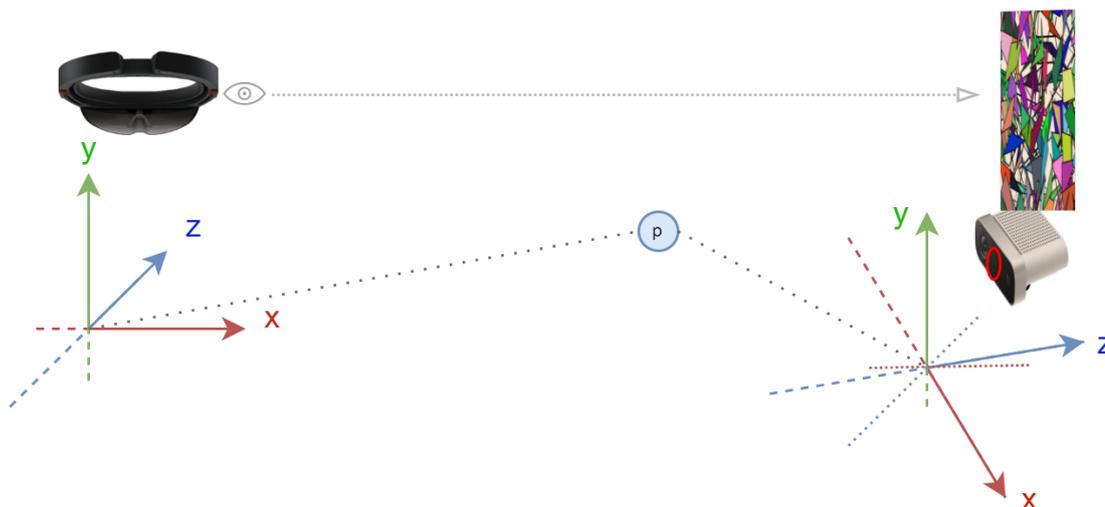


Figura 5.7: Creación del eje de coordenadas para el dispositivo de *tracking*

En la Figura 5.8 se puede observar el esquema de este módulo del sistema. Para recibir la posición de las articulaciones se dispondrá de un hilo que estará siempre a la escucha. Igual ocurre con el detector de marcas de *Vuforia*, cuando cualquiera de los dos reciba un evento, se le comunicará al módulo de *tracking* que se encargará del proceso comentado en el párrafo anterior.

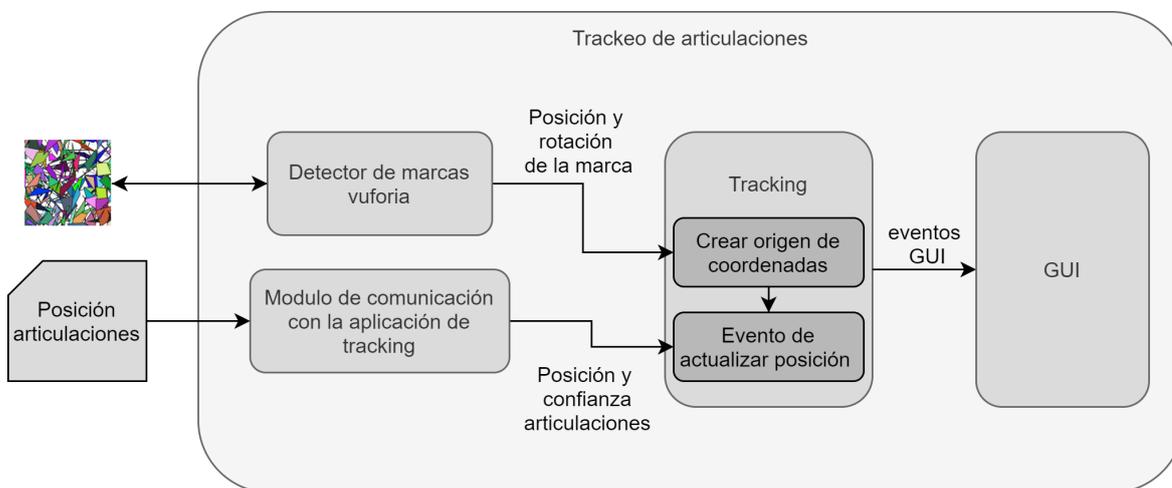


Figura 5.8: Vista general del *trackeo* de manos

Algoritmo 5.1: Algoritmo de definición de ejercicios

```

Datos      : mano, separación_puntos, dist_clave
Resultado: Ejercicio
1 puntos ← [];
2 puntos_aux ← [];
3 indice ← 0;
4 while grabando do
5     posicion_mano ← posicionDeMano(mano);
6     if nivel de confianza de la articulación es óptimo then
7         if indice == 0 then
8             puntos_aux[indice] ← posicion_mano;
9             puntos[indice] ← transformar(posicion_mano);
10        else
11            posicion_anterior ← puntos_aux[indice];
12            distancia ← abs(posicion_anterior – posicion_mano);
13            if distancia >= separacin_puntos then
14                puntos[indice] ← transformar(posicion_mano);
15                puntos_aux[indice] ← posicion_mano;
16            end
17        end
18        indice ← indice + 1;
19    end
20 end
21 Ejercicio(mano,puntos,dist_clave)

```

5.3.3. Definición de ejercicios

Uno de los objetivos de *PhyRe UP!* es que el terapeuta pueda definir ejercicios a través de la aplicación de RA, definiendo así un plan de rehabilitación. Los ejercicios están diseñados como un juego en el que el paciente tendrá que ir siguiendo un camino formado por puntos y aros hasta pasar por todos ellos. Mediante el seguimiento de esos puntos, el paciente hará un movimiento adecuado para recuperar la movilidad de forma progresiva. El terapeuta haciendo uso de la aplicación de RA podrá definir estos ejercicios y, para ello, tendrá que indicar unos parámetros y realizar la grabación del mismo.

Para la definición de ejercicios se va a seguir el siguiente Algoritmo 5.1. Antes de empezar, el terapeuta, a través de la interfaz gráfica, indica los parámetros del ejercicio: a) mano con la que se realiza el ejercicio, b) distancia en centímetros entre los puntos y cada cuántos puntos poner un punto clave (este punto clave se representará como un aro). Para iniciar el algoritmo se hará uso de un comando por voz, para que, de esta forma, el terapeuta tenga la mano libre para poder iniciar el ejercicio. Mientras se está grabando se obtiene la posición de la mano indicada para ese ejercicio. Solo se tiene en cuenta la información de la articulación si su nivel de confianza es el máximo, ya que, si el nivel de confianza es menor, no se puede concretar con exactitud la información de la posición de la articulación. Si la distancia entre

el punto anterior y el nuevo es mayor o igual a la distancia entre puntos que ha definido, se guardará el punto. Cuando el usuario invoque otro comando de voz se parará la grabación, el terapeuta introducirá los últimos parámetros: c) si el ejercicio es de ida y vuelta, y d) el nombre del ejercicio. A partir de esta información se podrá finalizar la creación del ejercicio. Para guardar se realiza una petición al *endpoint* de la *API Rest* enviando el ejercicio en formato *JSON*.

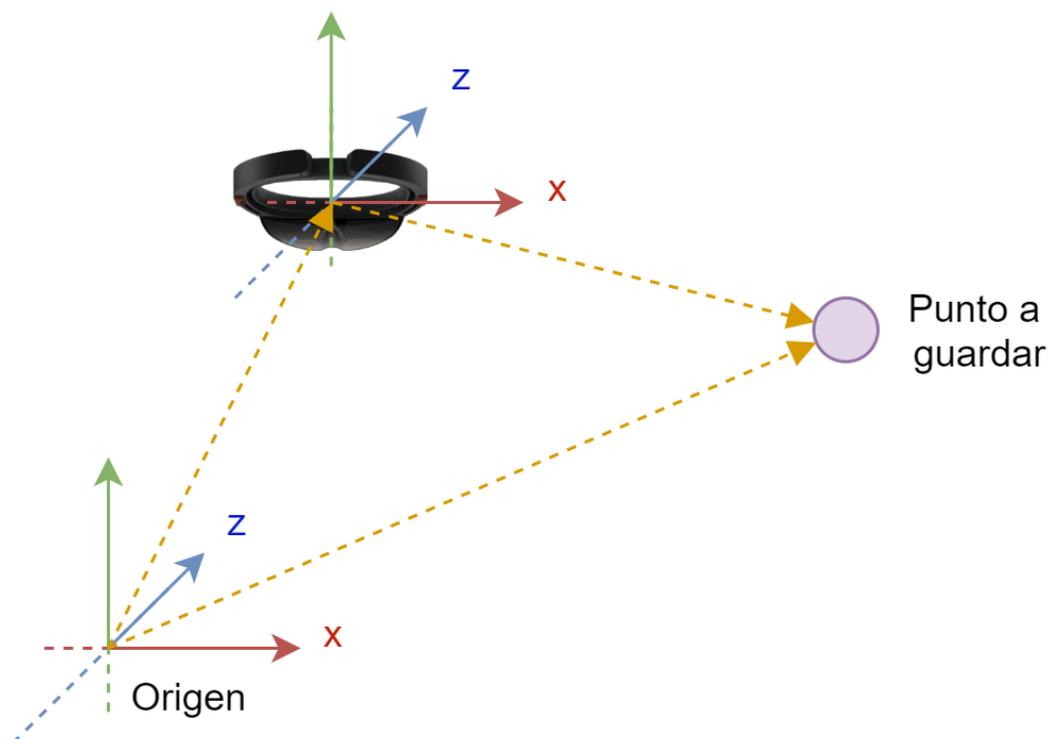


Figura 5.9: Posición del punto guardada para el ejercicio

Se puede observar en el Algoritmo 5.1 que se utilizan dos listas de puntos. Como se ha explicado en la Sección 5.3.2 al iniciar la aplicación se crea el origen de coordenadas en el punto inicial donde hayamos iniciado la aplicación. Si después, el usuario se mueve, el origen de coordenadas sigue anclado a ese punto. Por esta razón, no se pueden guardar los puntos del ejercicio con respecto al origen de coordenadas, ya que si un usuario inicia la aplicación en un lugar diferente del que luego realizará los ejercicios, estos ejercicios no estarán en la posición donde el terapeuta los ha creado. Para solucionar este problema, los puntos del ejercicio que se guardarán se tomarán con la referencia de la posición del dispositivo *HoloLens* como se puede observar en la Figura 5.9. De esta forma, al tomar como referencia el dispositivo *HoloLens* se evita el uso de otras marcas que creen dependencia del medio físico. El terapeuta define cada cuantos centímetros guardar un punto, para realizar esta comprobación se hace uso de la distancia respecto al origen de coordenadas y no desde el dispositivo *HoloLens*. Cuando el paciente realice el ejercicio se le mostrará como referencia su posición, y no el origen de coordenadas de su espacio virtual.

5.3.4. Definición y asignación de rutinas

Los ejercicios de rehabilitación que van a realizar los pacientes (también llamados rutinas) constan de un ejercicio (definido anteriormente) y unas repeticiones. El paciente realizará el ejercicio «n» veces. Para la creación de estas rutinas, el terapeuta, a través de la interfaz puede seleccionar un ejercicio de los que ha creado, indicar las repeticiones y el tiempo máximo que tiene el paciente para realizar todas las repeticiones. De esta forma se define una rutina. El terapeuta a través de la aplicación de RA puede asignar estas rutinas a los pacientes, constituyendo así un plan de rehabilitación a medida que se ajuste al problema concreto del paciente.

5.4. APLICACIÓN DE RA PARA LA REHABILITACIÓN DEL PACIENTE

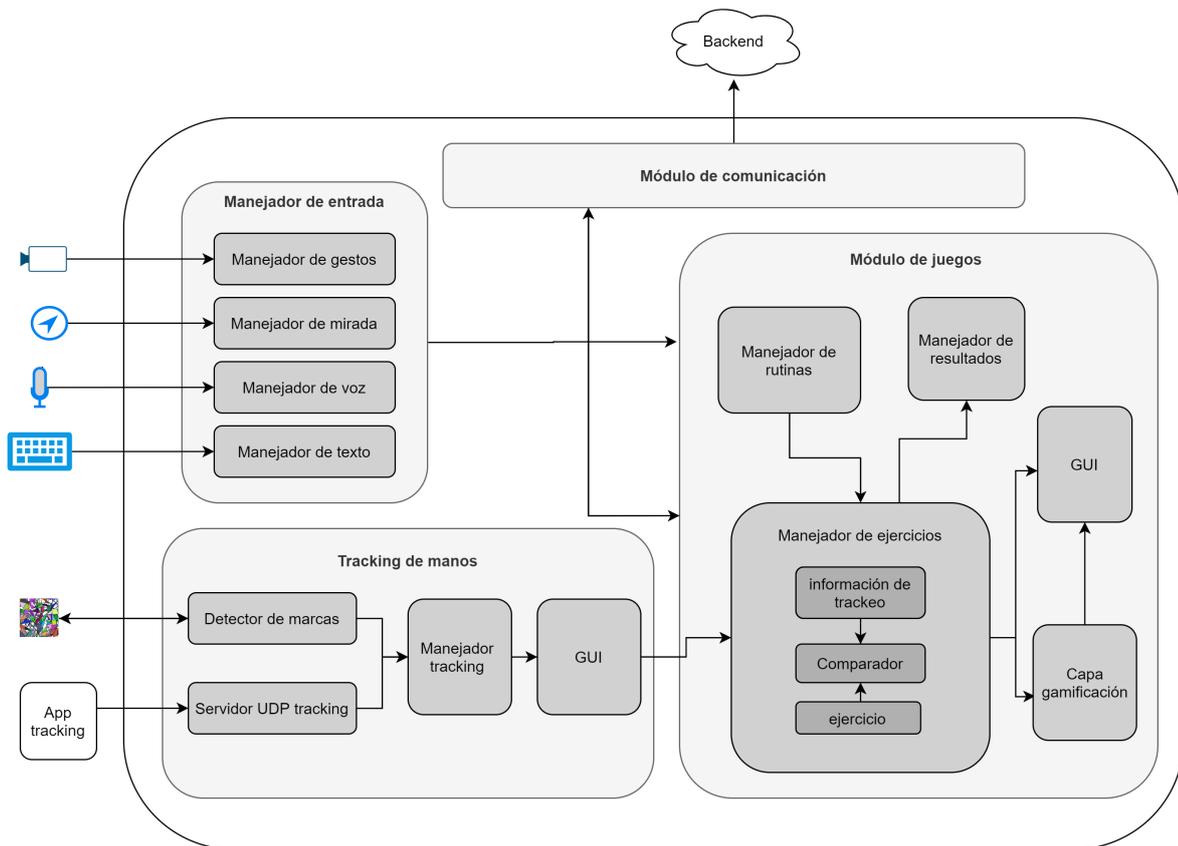


Figura 5.10: Vista general de la aplicación de RA para el paciente

El otro subsistema de *PhyRe UP!* es la aplicación de RA para el paciente. En ella, el paciente va a realizar los ejercicios en forma de juegos. En la Figura 5.10 se muestra una vista general de la arquitectura del subsistema. El módulo de comunicación, manejador de entrada y *tracking* de manos son los mismos que en la aplicación del terapeuta. Esta es una clara ventaja de una aplicación modular. Debido a que han sido descritos en secciones anteriores, no se volverá a incluir dicha descripción en esta sección (ver Sección 5.3).

5.4.1. Representación del ejercicio de rehabilitación

Para representar los puntos del ejercicio, que marcan el movimiento que debe reproducir el paciente, se ha usado el mismo principio que en la definición de ejercicios. El punto de referencia sobre el que se «dibujan» los puntos del ejercicio es el eje creado en la posición de las *HoloLens*. Esto permite no depender de ningún objeto o marca física en la estancia donde el paciente realiza la rehabilitación.

5.4.2. Realización de los ejercicios

El paciente tiene una lista con todos los ejercicios que tiene asignados, organizados como rutinas. Cada rutina consta de un ejercicio y unas repeticiones como se ha explicado anteriormente. Una vez realizadas todas las repeticiones (si el paciente no puede continuar puede parar el ejercicio con un comando de voz) recibe una evaluación sobre la exactitud con la que ha realizado el ejercicio. La evaluación del ejercicio la realizará el *backend* comprobando los nuevos datos del ejercicio con los anteriores. Cada ejercicio tiene un tiempo en el que debería terminarlo. Si el paciente tarda mas de lo indicado para realizar el ejercicio, se le mostrará un mensaje para indicarle que debe realizarlo en menos tiempo.

Cada ejercicio consta de unas esferas y unos aros que marcan la trayectoria del ejercicio que el paciente debe hacer. Los puntos muestran el inicio de la ruta que debe seguir el paciente, cuando llegue a los aros, aparecerán los siguientes puntos por donde deberá pasar. Con la intención de guiar al paciente por los puntos por los que debe pasar, estos se marcan cambiando de color según el orden. Los eventos que se desencadenan durante la realización del ejercicio son explicados en la Sección-5.5.

Esto puede verse en el Algoritmo 5.2. Al igual que ocurre con la definición de ejercicios, solo se evaluará la posición de la articulación del paciente si su nivel de confianza es el máximo. El paciente puede saber cuál es el nivel de confianza en cualquier momento a través de un cursor en forma de esfera que se sitúa encima de la mano usada para el ejercicio. Dicho cursor cambiará de color dependiendo del nivel de confianza. Si una vez completado el ejercicio, éste es de ida y vuelta vuelve a dibujarse el camino salvo que a la inversa.

Algoritmo 5.2: Algoritmo de realización de ejercicios de rehabilitación

```

Datos :Ejercicio
Resultado:
1 puntos ← ejercicio.puntos;
2 finalizado ← false;
3 indice ← 0;
4 acumulador_fallos = 0;
5 while no finalizado do
6   posicion_mano ← posicionDeMano(mano);
7   if nivel de confianza de la articulación es óptimo then
8     distancia ← abs(posicion_mano, puntos[indice]);
9     if distancia ≤ 0,01 then
10      guardar resultados de tiempo;
11      lanzar evento +1;
12      acumulador_fallos = 0;
13      if esUnAro(puntos[indice]) then
14        lanzar evento: aro completado;
15        renderizar los siguientes puntos;
16      else
17        lanzar evento: punto completado;
18      end
19      indicar siguiente punto;
20      indice ← indice + 1;
21    else
22      if pasa_otro_punto then
23        acumulador_fallos ← acumulador_fallos + 1;
24        if acumulador_fallos > numero then
25          lanzar evento de fallo;
26        end
27        guardar información de fallos;
28      end
29    end
30    if indice => numero_de_puntos then
31      if ida_y_vuelta(ejercicio) then
32        dibujar el ejercicio a la inversa;
33      else
34        finalizado ← true;
35      end
36    end
37  end
38 end

```

5.5. GAMIFICACIÓN

Con el fin de aumentar la motivación del paciente con su rehabilitación y así reducir la posibilidad de que el paciente abandone los ejercicios de rehabilitación se han usado técnicas de gamificación. El ejercicio se le presenta al paciente como un juego, donde deberá seguir una trayectoria (ejercicio), marcada por una serie de esferas y aros como se ha comentado en la sección anterior. Dependiendo de las acciones del paciente recibirá un *feedback* correspondiente. Esto se ha comentado en las Secciones 5.3.3, 5.4.2 y mostrado en la 6.1.3, por ello esta sección será mas breve.

En la Tabla 5.3 se pueden observar las acciones del paciente y los eventos que desencadenan. Cuando el paciente pase por un aro o punto, se activa una animación en el cursor que marca la posición de la mano, a la vez que aumenta el contador de puntos, se muestra una partícula con el texto «+1» y se marca el siguiente punto por el que el paciente debe pasar. Estos eventos son comunes tanto para los puntos (esferas) como para los aros, pero el evento sonoro es diferente para cada tipo. Si el paciente se equivoca al realizar el ejercicio se desencadena dos eventos, uno que hará parpadear el punto siguiente por el que debe pasar, y un evento sonoro que le ayude a saber que se está equivocando.

Acción	Evento	Tipo de feedback
Pasar por un aro/punto	Animación cursor de mano	Visual
	Mostrar «+1» punto	Visual
	Aumento de contador	Visual
Pasar por un aro	Sonido de aro	Sonoro
	Mostrar siguientes puntos	Visual
Pasar por un punto	Sonido de punto	Sonoro
Equivocarse de punto	Parpadeo de punto siguiente	Visual
	Sonido de error	Sonoro

Tabla 5.3: Acciones y eventos desencadenados

De esta manera los pacientes visualizarán las trayectorias de los ejercicios lo que consigue que puedan realizar el ejercicio de forma autónoma con mayor exactitud que realizándolo sin el sistema. Además, al presentar el ejercicio en forma de juego según diversos estudios ya comentados en la Sección-3.3, se aumenta la motivación del paciente. En este sistema se pretende el mismo efecto de aumentar la motivación del paciente al mostrar los ejercicios en forma de juegos.

Cuando el paciente está realizando el ejercicio y pase por los puntos correctos, recibirá un *feedback* visual indicándole que ha obtenido un punto y un *feedback* sonoro de recompensa para que, de esta forma el paciente pueda saber que está realizando el ejercicio correctamente. Al finalizar el ejercicio, el paciente también recibirá un mensaje con su evaluación con pasados ejercicios indicándole si ha mejorado o lo que tiene que mejorar. Al hacer uso de estas técnicas se pretende mantener al paciente motivado durante su periodo de rehabilitación.

5.6. BACKEND

Las aplicaciones que necesitan lidiar con datos anteriores para realizar un análisis o ver el progreso necesitan hacer uso de un almacenamiento de esa información. Este es el caso de los resultados de los ejercicios que realiza el paciente para calcular su evaluación. Además, se pretende que el terapeuta pueda definir ejercicios y rutinas y asignárselos al paciente sin que sea necesaria un actualizado manual de los datos en el dispositivo del paciente. Para ello, en esta sección se va a explicar las decisiones que se han llevado a cabo para poder realizar esta persistencia de los datos.

5.6.1. Vista general

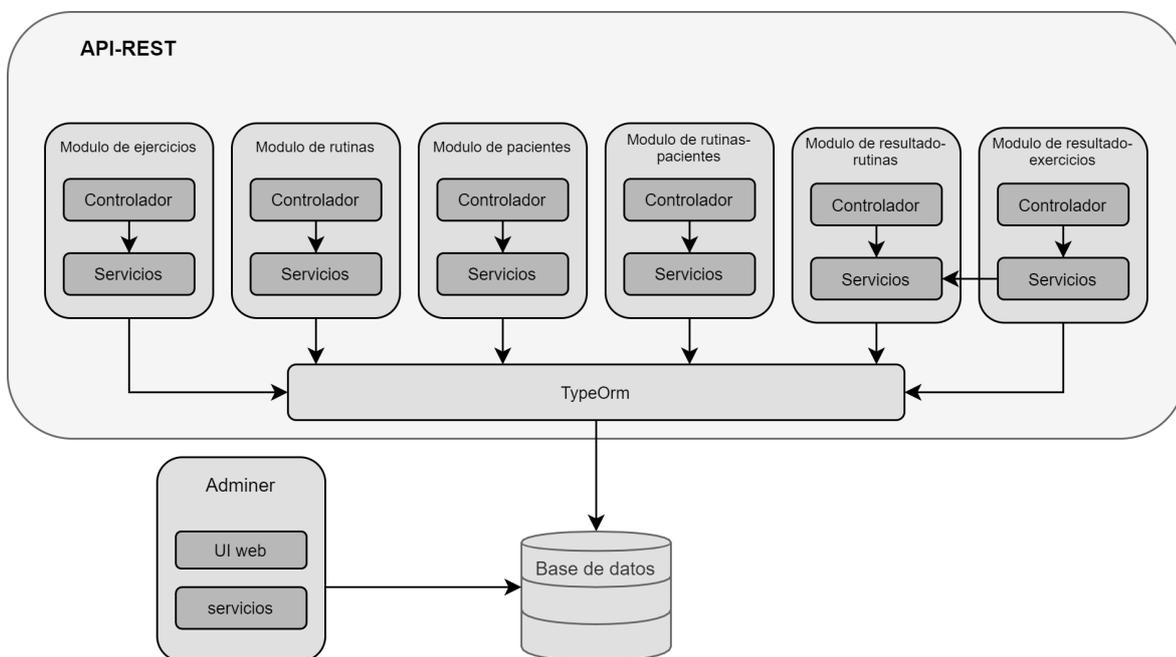


Figura 5.11: Vista general del backend

Como se puede observar en la Figura 5.11 el *backend* cuenta con tres sistemas. El servidor con la *API Rest* con la que se comunica la aplicación de RA para la creación y guardado de rutinas, ejercicios y resultados, así como la asignación de las rutinas a los pacientes. También es el encargado de devolver una evaluación de los ejercicios hechos por un paciente y de recuperar los datos. A su vez, otro de los sistemas será la base de datos con su sistema gestor de bases de datos, que en este caso ha sido elegido *PostgreSQL* cuyas ventajas serán comentadas en la siguiente sección. Siendo, el tercer y último sistema, *adminer*, para el manejo y administración de la base de datos mediante una interfaz web.

5.6.2. Base de datos

Para la persistencia del sistema *PhyRe UP!* se ha decidido hacer uso de una base de datos relacional. Como sistema gestor de la base de datos se ha elegido *postgreSQL* frente a otros

SGBD como *MySQL* o *OracleDB* debido a sus ventajas. *PostgreSQL* [14] cuenta con una alta concurrencia debido al uso del sistema denominado *acceso concurrente multiversión* o MVCC [4], sistema que permite a un proceso escribir en una tabla, mientras otros procesos acceden a la misma tabla sin necesidad de bloqueos y obteniendo una visión consistente de los datos. Otra de las ventajas de este SGBD es la posibilidad de guardar vectores y de poder definir tipos de datos propios. El último aspecto a destacar de *PostgreSQL* frente a otros, es su licencia. *PostgreSQL* está lanzado bajo la licencia *PostgreSQL License* [16], es una licencia de código abierto similar a BSD.

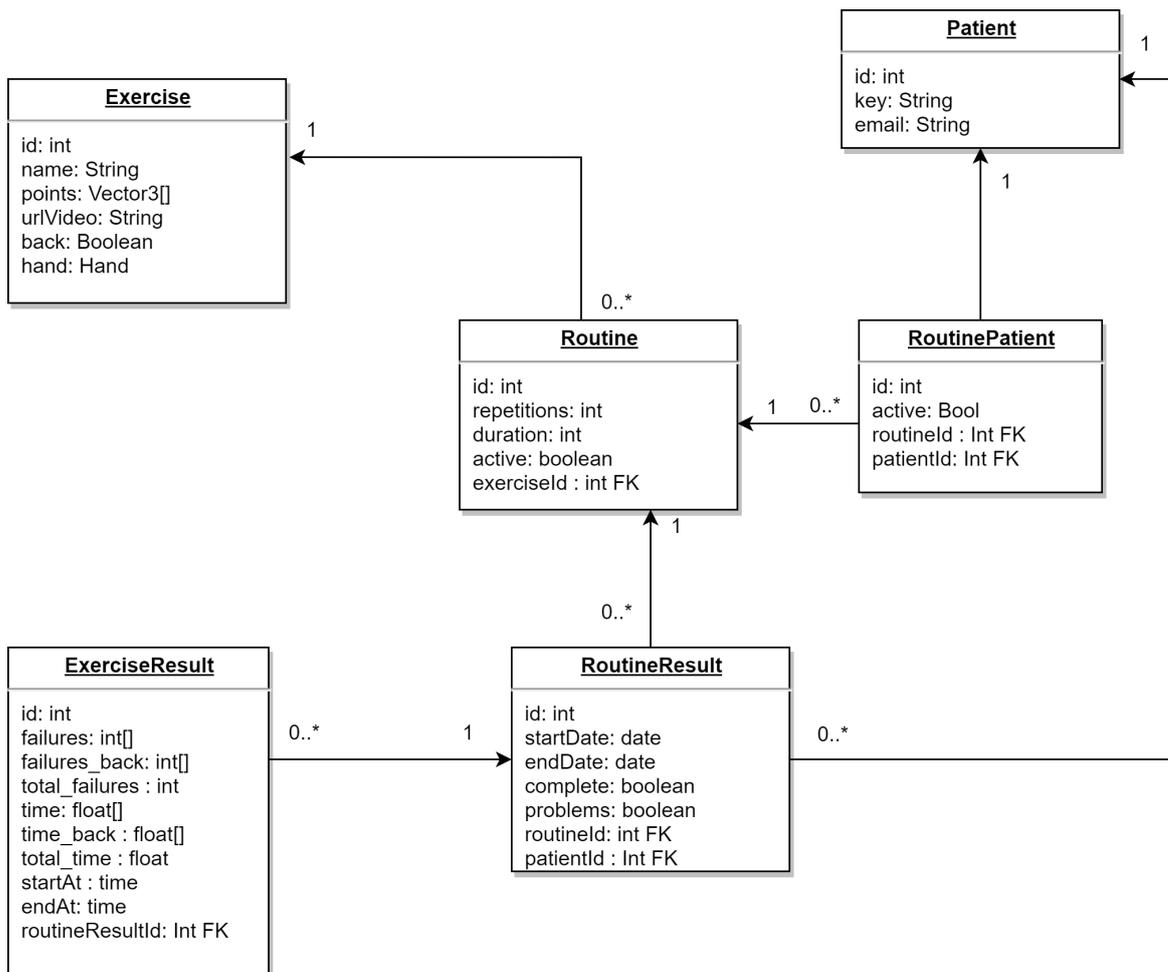


Figura 5.12: Esquema de la base de datos

El esquema diseñado para la base de datos se puede ver en la Figura 5.12. Se puede observar como hay dos tipos de datos que se han definido: *Hand*, un enumerado que representa la mano izquierda y derecha, y *Vector3* que representa un punto en el espacio (coordenadas x,y,z). La base de datos está formada por seis tablas. La tabla *Patient* que guarda la información referente al paciente, correo electrónico y una clave que será la que introduzca el paciente para obtener sus ejercicios. Los ejercicios se representan con una lista de puntos en el espacio, un nombre, la mano con la que se debe realizar los ejercicios y si el ejercicio es de ida y vuelta. Un paciente no realiza un ejercicio como tal, si no una *rutina*. Una rutina está compuesta por

un ejercicio, la cantidad de repeticiones que debe realizar y el tiempo máximo, de esta forma se puede reutilizar un ejercicio para hacer diferentes repeticiones. De ahí que la relación entre paciente y rutina se realice a través de la tabla *RoutinePatient*, ya que, de esta forma, una rutina se puede reutilizar para más pacientes.

El sistema deberá guardar los resultados de los ejercicios puesto que será necesario para saber si el progreso del paciente se está realizando acorde a lo esperado. Para ello se ha decidido guardar los resultados diferenciando entre resultados de rutina y resultados de ejercicios. Como se ha explicado anteriormente, una rutina tiene un ejercicio y varias repeticiones que deberán hacerse. Por tanto, cada repetición se guardará en la tabla *ExerciseResult*, que a su vez tendrán una relación con el resultado de la rutina en *RoutineResult*. Siendo también necesaria una relación entre *RoutineResult* y el paciente. Es importante destacar que para los datos de tipo fecha se ha elegido guardarlos como cadenas de texto siguiendo el estándar ISO 8601³, por lo que las fechas tendrán la forma: *YYYY-MM-DDThh:mm:ss.sTZD*.

Listado 5.3: Ejemplo de la entidad de *RoutineResult*

```

1
2 @Entity ()
3 export class RoutineResult {
4
5     @PrimaryGeneratedColumn ()
6     id: number
7
8     @Column ()
9     @IsString ()
10    startDate: string;
11
12    @Column ()
13    @IsString ()
14    endDate: string;
15
16    @Column ()
17    @IsBoolean ()
18    complete: boolean;
19
20    @Column ()
21    @IsBoolean ()
22    problems: boolean;
23
24    @ManyToOne (type => Routine, routine => routine.routineResults)
25    routine: Routine;
26
27    @ManyToOne (type => Patient, patient => patient.routineResults)
28    patient: Patient;
29
30    @OneToMany (type => ExerciseResult, exerciseResult => ←
31                ↪ exerciseResult.routineResult)
32    exerciseResult: ExerciseResult [];

```

La creación de las tablas de la base de datos se ha realizado a través de *TypeOrm* [45]. *TypeORM* es un *ORM* para *NodeJS*. El mapeo objeto-relacional, o por sus siglas en inglés *ORM* [30], es una técnica de programación que permite convertir datos entre los objetos

³<https://www.iso.org/iso-8601-date-and-time-format.html>

de la programación orientada a objetos y una base de datos relacional usada como motor de persistencia del sistema. Por lo que para la creación de las tablas de la base de datos solo se ha necesitado la definición del objeto como una clase en *TypeScript* y hacer uso de los decoradores de *TypeOrm*. A este objeto que modela una tabla de la base de datos se le denomina *entidad*.

En el Listado 5.3 se puede observar un ejemplo de entidad. En esta se define la tabla *RoutineResult*. Se puede ver como el id se indica como auto-generado y los demás campos como columnas. Además, se indican las relaciones bien sean entrantes o salientes con los decoradores *OneToMany* o *ManyToOne*. También se hace uso de decoradores para validar los tipos de datos que lleguen en las operaciones *CRUD* a la *API Rest*.

5.6.3. Contenedores

Para el uso y despliegue de esta aplicación se ha optado por hacer uso de *Docker*. *Docker* es una plataforma *software* que permite empaquetar *software* en lo que se conoce como «contenedores». Estos contenedores incluyen todo lo necesario para que el *software* se ejecute (bibliotecas, herramientas, código, etc). Las máquinas virtuales también son usadas para esta función. Las máquinas virtuales necesitan hacer uso de un hipervisor (ver Figura 5.13), que es una plataforma que permite hacer uso de la virtualización. Este hipervisor va sobre el sistema operativo de la máquina (también puede ir directamente sobre la infraestructura), sobre este hipervisor se instalará el sistema operativo de la máquina virtual, junto a las bibliotecas y herramientas necesarias para el funcionamiento de esa aplicación. Con los contenedores de *Docker* desaparece el sistema operativo huésped y el hipervisor se cambia por *Docker Engine*. Aunque parezca que el cambio es mínimo, las diferencias son enormes. *Docker Engine* se encarga de lanzar y gestionar los contenedores pero en lugar de dedicar diferentes recursos *hardware* a cada contenedor, comparte estos recursos entre los contenedores optimizando su uso, siendo también posible reutilizar las bibliotecas y herramientas (conocidas como imágenes) entre varias aplicaciones [24].

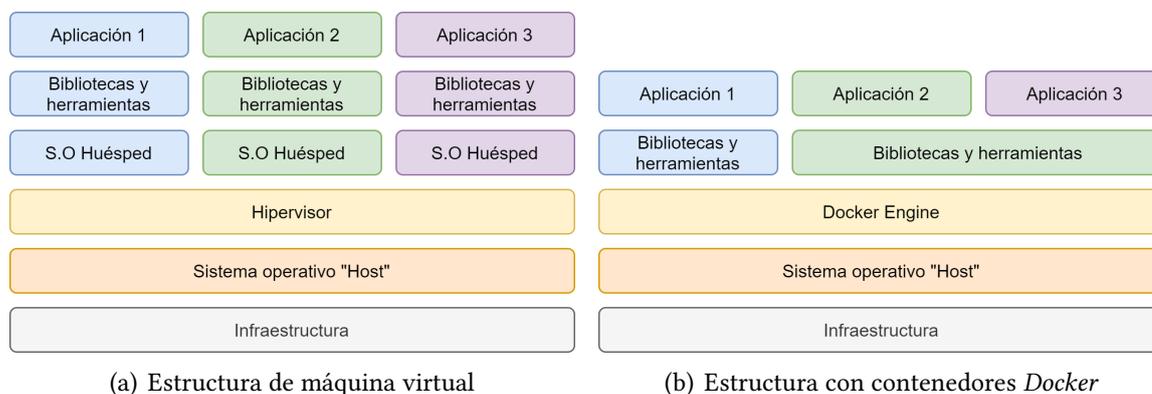


Figura 5.13: Comparación entre *Docker* y máquinas virtuales

Por estas ventajas se ha decidido hacer uso de *Docker* con lo que se ha creado un contenedor para cada una de las tres partes que conforman el *backend*. Los contenedores contienen lo siguiente:

- *Adminer*. Una herramienta que va a permitir acceder a la base de datos para labores de mantenimiento o administración.
- *PostgreSQL*. Sistema gestor de la base de datos como se ha explicado en la subsección anterior.
- *Servidor de Node.js*. Servidor usando el *framework* de *Node.js NestJS* para la construcción de la *API Rest*.

Para la creación del contenedor que contiene el servidor con la *API Rest* se ha creado una imagen con los componentes y dependencias necesarios. Para ello se ha hecho uso de *Dockerfile*. Se ha elegido usar el sistema operativo *Ubuntu:18.04* para la imagen del servidor. Se puede observar en el Listado 5.4 el *Dockerfile* usado para la creación de la imagen.

Listado 5.4: *Dockerfile* para la creación de la imagen del servidor de la *API Rest*

```
1 FROM ubuntu:18.04
2 RUN apt-get update
3 RUN apt-get install -y build-essential curl
4 RUN apt-get install -y nodejs
5 RUN curl -sL https://deb.nodesource.com/setup\_{12}.x | bash \-
6 RUN npm install -g ts-node typescript
```

Para la instanciación de los otros dos sistemas que componen el *backend* (base de datos y adminer) se ha hecho uso de imágenes ya creadas y *Docker Compose*. *Docker Compose* es una herramienta que permite definir y lanzar aplicaciones que hagan uso de múltiples contenedores *Docker*. Para ello se hace uso de un archivo *YAML* para la configuración de cada una de las máquinas. Tras eso, con un solo comando se podrán instanciar y levantar las máquinas que compongan el sistema de forma simultánea. En la Figura 5.14 se puede observar los tres diferentes contenedores y su comunicación entre ellos y el exterior.

5.6.4. API Rest

Para el acceso y manejo de los datos se ha decidido hacer uso de una *API Rest* [13]. Para la construcción de la API se ha decidido usar *NestJS*, que permite la abstracción del sistema en diferentes capas bien diferenciadas haciendo que el sistema sea escalable y fácil de administrar. Como se puede ver en la Figura 5.11 el servidor está dividido en diferentes módulos. Cada módulo tiene a su vez un controlador, una entidad y sus servicios, y son los encargados de las dependencias que necesitan dichos servicios y controladores. El controlador es el encargado de definir los *endpoints* de la API. Mientras que los servicios son los encargados de realizar las operaciones que demanden los controladores y los accesos a la base de datos.

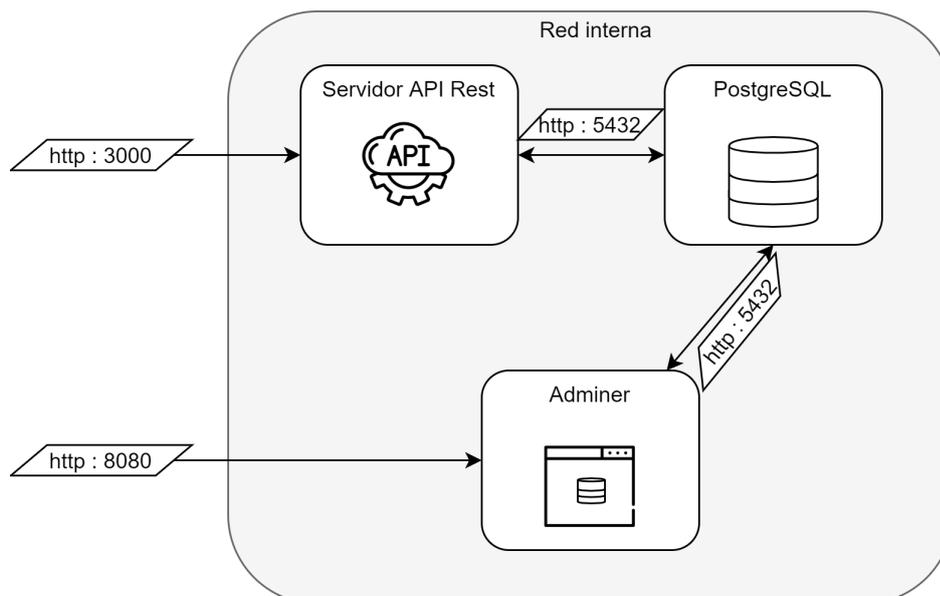


Figura 5.14: Comunicación entre los contenedores y el exterior

En el Listado 5.5 se puede ver la estructura del controlador del módulo *Routine*. La ruta de la URL del controlador es: `/routine`, por lo que los demás *endpoints* continuarán esa ruta. Este controlador simplificado cuenta con tres *endpoints*. El primero de ellos usa el método `http GET` y además solicita un parámetro por URL. El parámetro indica el identificador de la rutina que se va a pedir, también se puede observar el método `POST` que es usado para guardar una rutina. El campo para esta petición es un objeto rutina, de ahí que en las entidades se usen los validadores (Listado 5.3), para comprobar que las peticiones que lleven ese objeto estén bien construidas.

El último componente de los módulos de la API son los servicios. Esta clase es usada para realizar las consultas a la base de datos a través de los repositorios. Cada servicio tiene el repositorio asociado a su entidad y a través de él se hacen las operaciones sobre su entidad, es decir, sobre su tabla. Mediante el uso de *TypeOrm* no es necesario diseñar las consultas en SQL. *TypeOrm* se encargará de realizar la traducción de las instrucciones a lenguaje SQL que son ejecutadas en la base de datos. Un ejemplo de servicios se puede observar en el Listado 5.6 en el cual se muestra el servicio simplificado de la entidad *Routine*. En este servicio se puede observar una consulta a la base de datos para devolver una rutina por id y también se puede observar una consulta más compleja que hace uso de uniones (*joins*) para devolver todas las rutinas que pertenezcan a un paciente dado.

Listado 5.5: Ejemplo simplificado del controlador de *Routine*

```

1
2 @Entity()
3 @Controller("routine")
4 export class RoutineController {
5
6     constructor(
7         private readonly routineService: RoutineService
8     ) {}
9
10    @Get(":id")
11    async getRoutineById(@Param() params) {
12        const routineId = params.id;
13        return await this.routineService.getRoutineById(routineId);
14    }
15
16    @Get("patient/:patient")
17    async getRoutineByPatient(@Param() params) {
18        const patientKey = params.patient;
19        return await this.routineService.getRoutinesByPatientKey(patientKey);
20    }
21
22    @Post()
23    async saveRoutine(@Body() routine: Routine) {
24        return await this.routineService.register(routine);
25    }
26 }

```

Listado 5.6: Ejemplo simplificado del servicio de *Routine*

```

1
2 @Injectable()
3 export class RoutineService {
4
5     constructor(
6         @InjectRepository(Routine)
7         private readonly routineRespository: Repository<Routine>,
8     ) {}
9
10    async register(routine: Routine) {
11        return await this.routineRespository.save(routine);
12    }
13
14    async getRoutineById(id: number) {
15        return await this.routineRespository
16            .createQueryBuilder("routine")
17            .select(["routine"])
18            .where("id = :id", {id: id})
19            .getOne();
20    }
21
22    async getRoutinesByPatientKey(key: string) {
23        return await this.routineRespository
24            .createQueryBuilder("routine")
25            .select(["routine"])
26            .orderBy("routine.id", "ASC")
27            .innerJoin("routine_patient.routine", "routine")
28            .getMany();
29    }
30
31 }

```

CAPÍTULO 6

RESULTADOS

En este capítulo se van a mostrar, en primer lugar, los resultados del desarrollo del sistema. Primero se hará una exposición de la funcionalidad de las aplicaciones desarrolladas para el sistema «*PhyRe UP!*». En segundo lugar, se mostrará la aplicación de la metodología expuesta en el Capítulo-4 y los costes del sistema. Finalizando con la exposición de los resultados del sistema con un grupo de sujetos de prueba, aunque dada la excepcional situación de este año 2020, el grupo de prueba es reduce al círculo cercano del autor.

Para el control de versiones del código fuente se ha hecho uso de *Git*. Se han usado tres repositorios para las aplicaciones de RA, aplicación de *tracking* y *backend*. Desde el *readme* del repositorio principal se podrá acceder a los demás repositorios, además, desde este repositorio están los enlaces a los vídeos de demostración:

<https://github.com/DavidCarneros/PhyReUP-TFG-RA>

6.1. PHYSICAL REHABILITATION UP!

El nombre del sistema elegido para este sistema es *PhyRe UP!*, que proviene de las palabras anglosajonas *physical* (física) y *rehabilitation* (rehabilitación). En la Figura 6.1 se puede observar el logotipo del sistema. Para desarrollo del sistema se han implementado las cuatro aplicaciones que se han explicado en el Capítulo 5. En las siguientes sub-secciones se van a exponer los resultados del desarrollo de estas aplicaciones. En el anexo-A se puede observar una fotografía del sistema *PhyRe UP!*.



Figura 6.1: Logotipo del sistema PhyRe UP!

6.1.1. Tracking de manos

El *tracking* de manos se usa tanto en la aplicación del terapeuta como en la del paciente. Para ello, la aplicación de *tracking* desarrollada envía la información de las manos a la aplicación de RA que se esté usando.

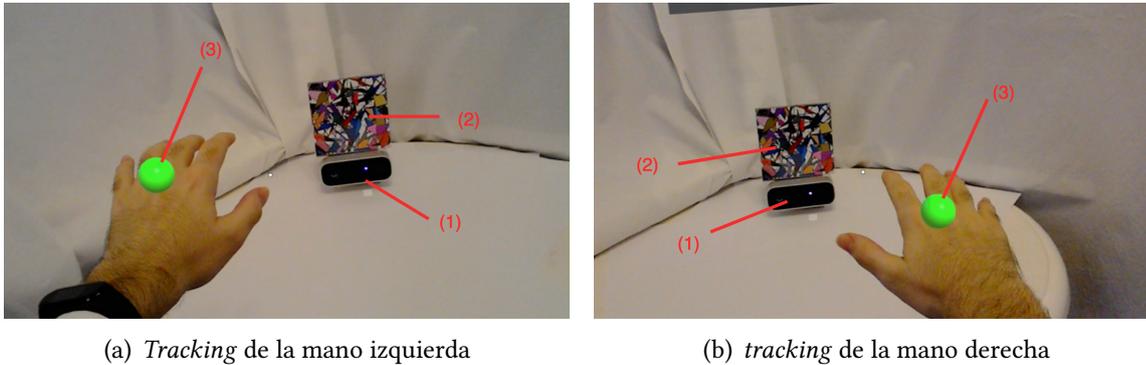


Figura 6.2: Tracking de mano vista desde la aplicación de RA

En la Figura 6.2 se observa el *tracking* de las mano desde la aplicación de RA. En esta Figura se distingue el dispositivo de *tracking* (1), que está conectado al *PC Host* enviando los datos de las articulaciones al dispositivo de RA (*HoloLens*) que usa el paciente o terapeuta. Para solventar el problema de localización del dispositivo de *tracking* se hace uso de una marca (2) como se ha expuesto en el Capítulo-5. Como *feedback* que recibe el usuario de la aplicación, se observa un puntero esférico (3) en la mano del usuario. Este puntero estará siempre siguiendo la articulación y presentará diferentes colores dependiendo del nivel de confianza obtenido por la aplicación de *tracking*. Estos niveles son: a) verde, mayor nivel de confianza de la articulación; b) amarillo, la posición de la articulación no puede ser obtenida con exactitud y c) rojo, la posición de la articulación no puede ser conocida.

6.1.2. Aplicación del terapeuta

La aplicación del terapeuta ha sido desarrollada siguiendo la arquitectura expuesta en el Capítulo- 5. A continuación, se va a mostrar el resultado del desarrollo de esta aplicación de RA para el dispositivo *HoloLens*.

En la Figura 6.3 se observa el menú principal que ve el terapeuta con el que podrá navegar por la aplicación. En este menú se observa el contenedor (1) sobre el que todos menús se van a mostrar, el usuario puede crear un ejercicio (3), crear y asignar rutinas para los pacientes (4) y visualizar los ejercicios y los progresos de los pacientes (2). Estos botones se pueden pulsar mirando hacia el botón y haciendo el gesto «click» que reconocerá el dispositivo *HoloLens*.

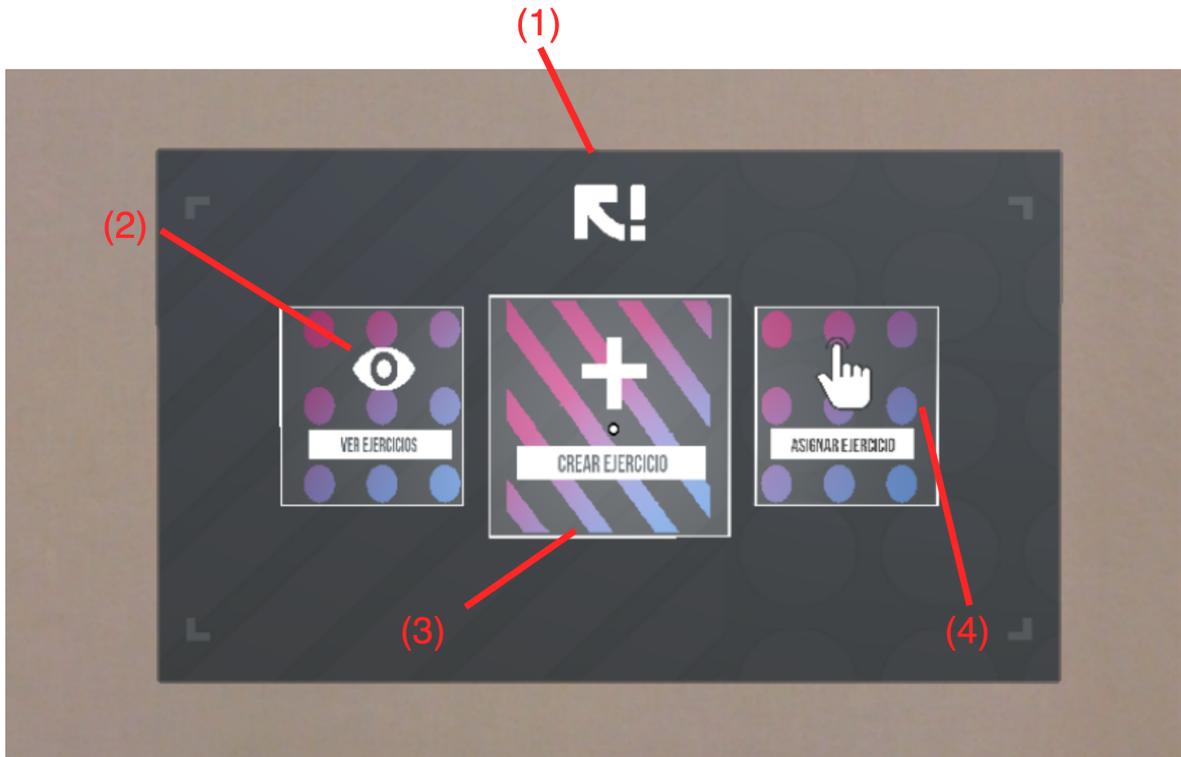


Figura 6.3: Menú principal de la aplicación RA para el terapeuta

Definición de ejercicios

El terapeuta, a través de la aplicación de RA, puede definir un ejercicio. La definición de ejercicios cuenta con cinco fases. La primera fase es el selector de la mano para la cual se va a definir el ejercicio. En la Figura 6.4(a) se observa el menú para la selección de mano que se compone de dos botones (1). Cuando se seleccione uno de estos botones se pasa a la siguiente fase. Y el botón de volver (2) que vuelve al menú principal. En la siguiente fase 6.4(b) el terapeuta define los parámetros iniciales del ejercicio, estos parámetros se componen de la distancia en centímetros entre puntos (1) y sus botones para aumentar y disminuir su valor (2), y cada cuantos puntos colocar un punto clave (3) y sus botones de manejo (4).



(a) Selector de mano para el ejercicio

(b) Configuración de parámetros del ejercicio

Figura 6.4: Tracking de mano vista desde la aplicación de RA

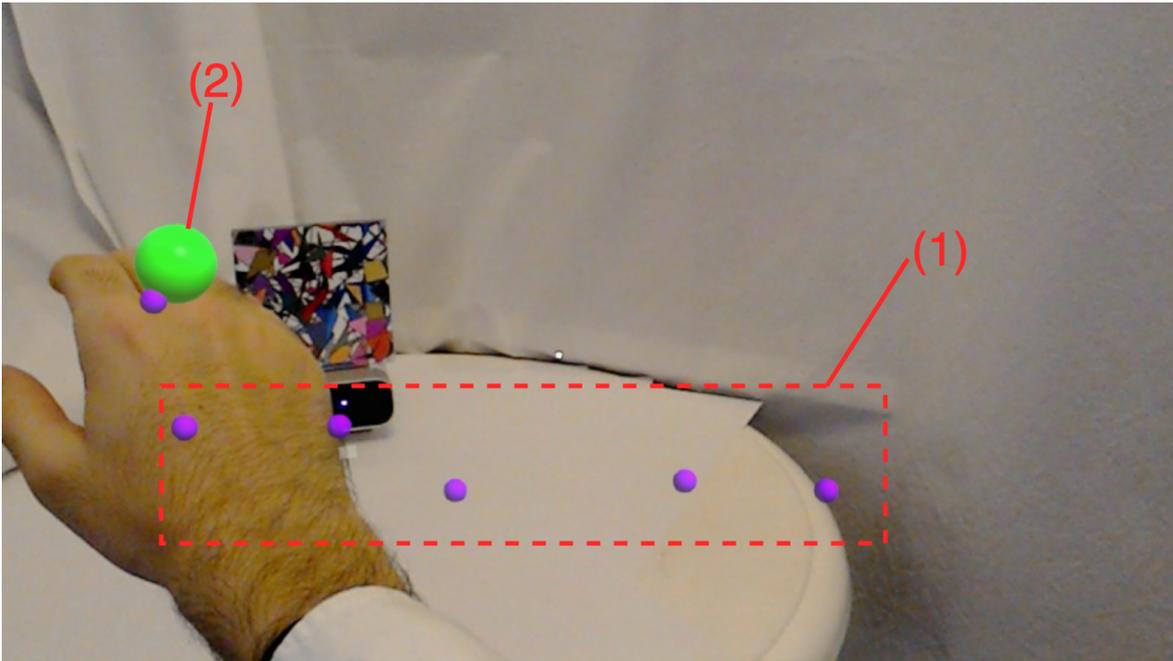
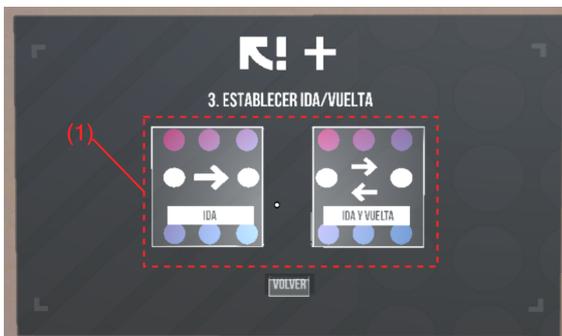


Figura 6.5: Captura de la grabación de un ejercicio

Tras seleccionar la mano y configurar los parámetros iniciales se puede comenzar a grabar el ejercicio usando los comandos por voz «start» y «stop». En la Figura 6.5 se observa el proceso de grabación del ejercicio, los puntos que se van guardando se van mostrando en pantalla (1), también puede observarse el puntero que indica la posición de la mano (2). Cuando se pare la grabación con el comando de voz se pasa a la siguiente fase de definición de ejercicios.

Las últimas fases de la definición de ejercicios consisten en definir los últimos parámetros del ejercicio. La siguiente fase define si el ejercicio es solo de ida o por el contrario, de ida y vuelta 6.6(a), para ello se presenta una colección de dos botones (1) al igual que en la selección de mano. En la última fase 6.6(b), el terapeuta definirá el nombre del ejercicio a través de la entrada de texto que aparecerá al pulsar sobre (1).



(a) Selector de ida / ida y vuelta



(b) Guardar ejercicio

Figura 6.6: Últimas fases de la definición de ejercicios

Entrada de texto

La entrada de texto en las aplicaciones de RA para el paciente y el terapeuta hará uso del teclado mostrado en la Figura 6.7. Para navegar por el teclado se hará uso del modo de interacción de mirada (*gaze*) y el gesto «click».

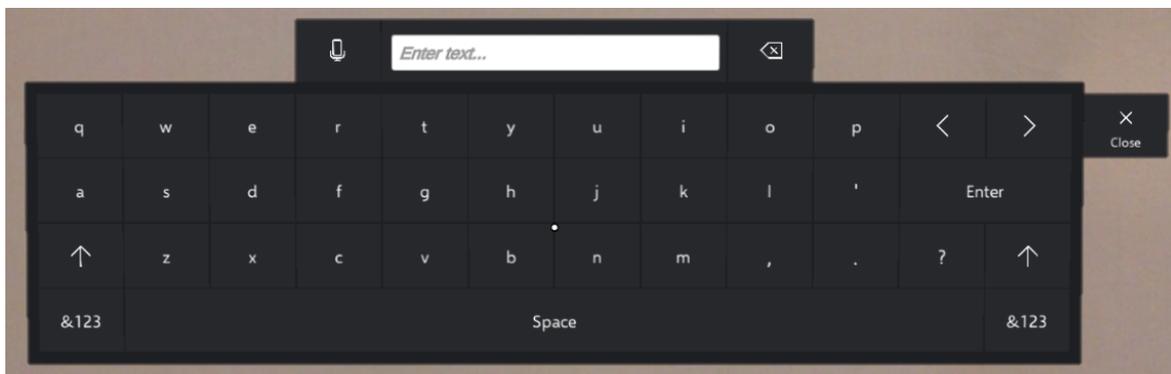
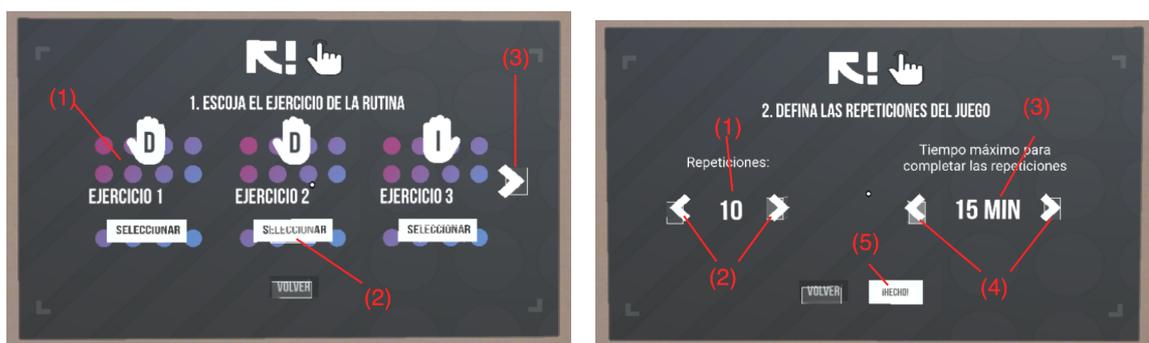


Figura 6.7: Entrada de texto para las aplicaciones de RA

Creación de rutinas

Navegando desde el menú principal (6.3) hacia asignar rutinas, el terapeuta tiene la opción de definir las rutinas y asignarlas a los pacientes. Como se ha comentado en el capítulo anterior, las rutinas que realizan los pacientes constan de un ejercicio con «n» repeticiones.

A través de la aplicación de RA, el terapeuta puede crear las rutinas para, posteriormente, asignarlas a los pacientes. En la Figura 6.8(a) se puede observar el primer paso para definir una rutina que es la selección del ejercicio. Para realizar esta tarea aparecen todos los ejercicios definidos en cuadros (1) indicando el nombre y la mano para la que se ha realizado el ejercicio. Una vez seleccionado el ejercicio (2) se pasa a la configuración de la rutina.



(a) Selector de ejercicio para la rutina

(b) Guardar ejercicio

Figura 6.8: Creación de rutinas

La última fase de definición de rutinas se muestra en la Figura 6.8(b), donde se observa la configuración de las repeticiones (1) a través de los botones (2) y la selección del tiempo máximo para realizar todas las repeticiones (3). Cuando acabe de realizar la configuración y pulse «hecho» (5), la rutina se creará.

Asignación de rutinas

Una vez definidas las rutinas, el terapeuta puede asignarlas a los pacientes. En la Figura 6.12 se observa el proceso de asignación de rutinas. En primer lugar, el terapeuta selecciona la rutina que previamente ha creado (Figura 6.9(a)) y por último introduce el identificador del paciente para realizar la asignación (ver Figura 6.6(b)).



(a) Selector de ejercicio para la rutina

(b) Guardar ejercicio

Figura 6.9: Asignación de rutinas

Visualizar ejercicios

El terapeuta puede visualizar los ejercicios que ha creado. Para ello tiene una lista con todos los ejercicios definidos como se puede ver en la Figura 6.10, en la que aparecen los ejercicios en un cuadro que indica el nombre y la mano para la que están definidos (1). El terapeuta selecciona el ejercicio que quiere ver (2) y se pasará a la representación del ejercicio.



Figura 6.10: Menú para elegir el ejercicio para ver

En la Figura 6.11 se observa la representación de un ejercicio, y éste aparece marcado por completo por el cuadro (1). El primer punto por el que el paciente comenzará a realizar el ejercicio se muestra en color amarillo y más grande que los demás (2). Los parámetros de separación entre puntos y cada cuántos puntos colocar un aro lo define el terapeuta como se ha comentado anteriormente. Como se ha explicado en el capítulo-5, un ejercicio está formado por unos puntos (4) y aros (3) que el paciente deberá ir recolectando.

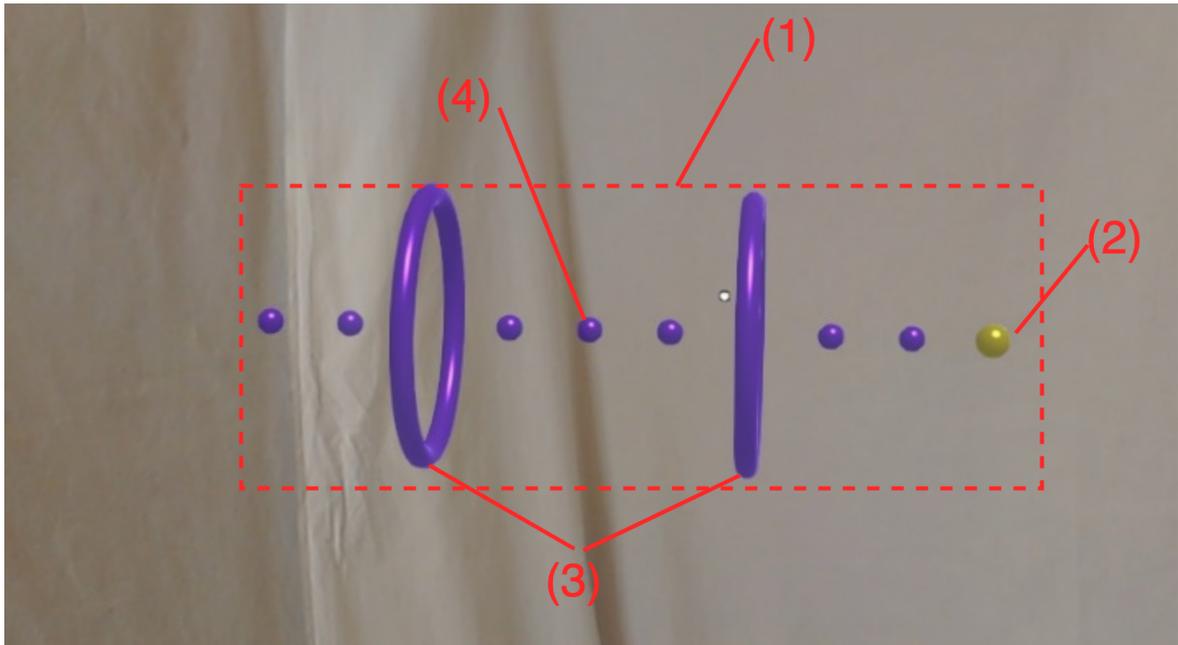
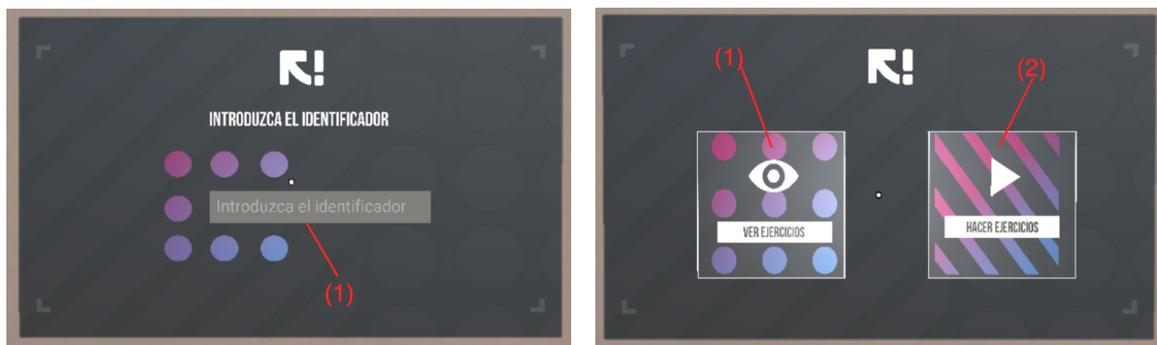


Figura 6.11: Visualización de ejercicio

6.1.3. Aplicación del paciente

El paciente al iniciar la aplicación encontrará una ventana en la que tendrá que introducir el identificador que le ha sido asignado (ver Figura 6.12(a)). Una vez introducido el identificador aparecerá el menú principal (ver Figura 6.12(b)). En el menú principal el paciente podrá ver las rutinas (1) y realizar las rutinas (2) que tenga asignados.



(a) Ventana inicial de la aplicación del paciente (b) Menú principal de la aplicación de RA del paciente

Figura 6.12: Primeras ventanas de la aplicación de RA del paciente

Realización de rutinas

Al paciente se le muestra una lista con las rutinas que tiene asignados (Figura 6.13). Las rutinas se representan indicando la mano con la que debe realizarse, el nombre y las repeticiones que han de hacerse (1). El paciente elige la rutina haciendo el gesto «click» en el botón del ejercicio que quiera hacer (2).

Antes de empezar con la realización de las rutinas, se le muestran al paciente mensajes de



Figura 6.13: Menú de selección de ejercicios a realizar

información como los que se pueden ver en la Figura 6.14. Si el paciente no sabe cómo hacer un ejercicio puede usar el comando de voz «*help*» que le mostrará un vídeo del ejercicio si el terapeuta ha definido uno. Si durante el ejercicio el paciente no puede continuar podrá parar usando el comando de voz «*stop*».



(a) Mensaje de explicación del ejercicio

(b) Mensaje de información del ejercicio

Figura 6.14: Mensajes de información mostradas antes de la realización del ejercicio

En la Figura 6.15 se muestra la vista del paciente cuando comience con el ejercicio. En ella se observa el punto por el que debe empezar (1), que irá pasando al siguiente conforme vaya recolectando los puntos. También puede observarse el marcador de puntos recolectados (2). Un detalle a destacar que diferencia la vista al realizar los ejercicios con la vista de ver el ejercicio es que solo se muestran los puntos hasta el siguiente aro, cuando el paciente termine de recolectar esos puntos y pase por el aro se le mostrarán los siguientes puntos. Se ha decidido esta forma de representar el ejercicio para no sobrecargar con demasiada información visual al paciente.

En la Figura 6.16(a) se distingue como el paciente ya ha recolectado dos puntos y se dispone a recolectar el siguiente marcado en amarillo. En la Figura 6.16(b) se muestra al paciente recolectando un punto. Cuando se recolecta un punto el puntero de la mano realiza una animación, además de emitir un sonido para que el paciente reciba un buen *feedback*. También se muestra una partícula «+1» para indicar que el punto ha sido conseguido.

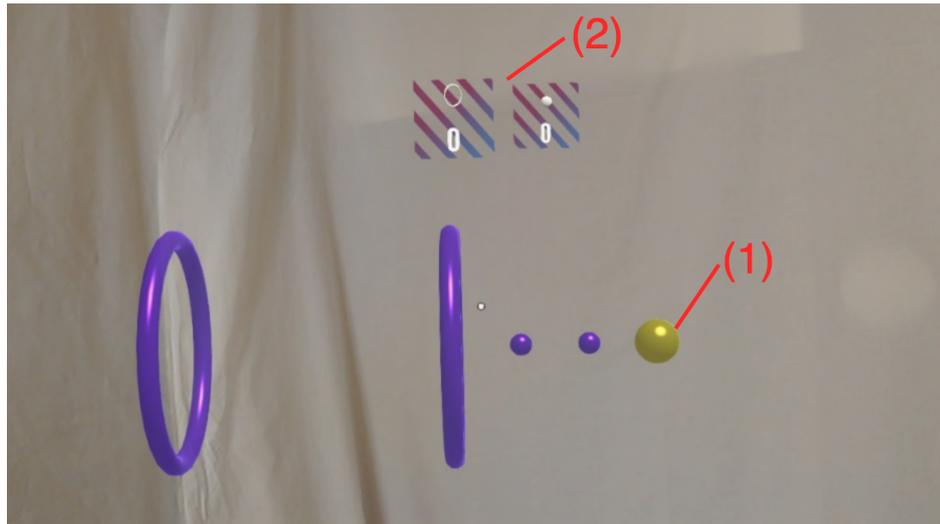
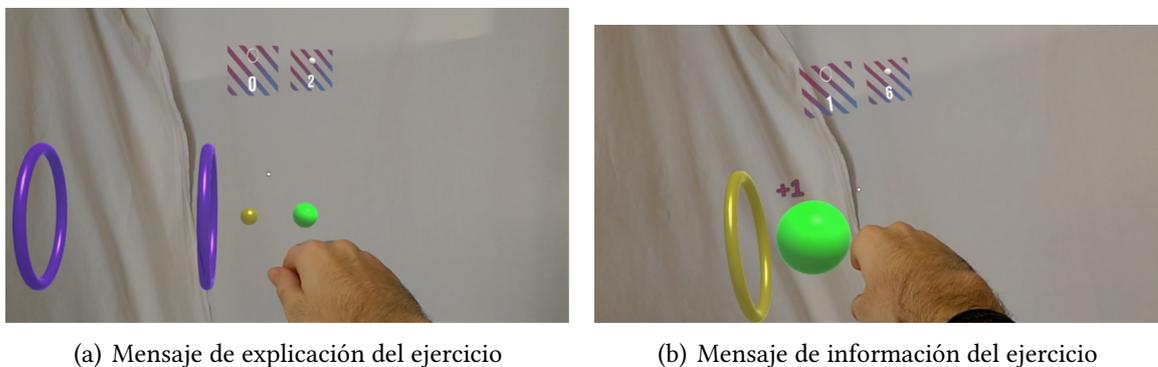


Figura 6.15: Vista del ejercicio antes de ser realizado por el paciente



(a) Mensaje de explicación del ejercicio

(b) Mensaje de información del ejercicio

Figura 6.16: Paciente realizando un ejercicio

Una vez el paciente llegue al final del ejercicio, en caso de que el ejercicio sea de ida y vuelta se le volverá a mostrar solo que en dirección inversa. En caso de ser solo de ida, habrá terminado una repetición. Se le muestra al paciente que va a comenzar otra repetición y volverá a hacer el juego. Tras terminar todas las repeticiones, o en caso de terminar por que no puede continuar se le muestra al paciente los resultados de la realización del ejercicio, indicándole si está mejorando o no sus resultados comparándolo con resultados anteriores.

6.1.4. Backend

Para el despliegue del sistema se ha elegido el servicio de computación en la nube *Microsoft Azure*. Para el sistema se han usado dos máquinas con las mismas características, máquinas de la serie *B2s estándar* que cuentan con dos núcleos de procesamiento y cuatro *Gibibytes* de memoria RAM. En una de las máquinas se ha desplegado la base de datos y en la otra la API y *adminer*. Se ha decidido esta arquitectura para facilitar el escalado de los sistemas que componen el *backend* por separado.

En la Figura 6.17 se observa la topología de la red. Las dos máquinas tienen dos interfaces de red, una conectada a una intranet y otra a internet. La comunicación entre las dos máquinas se realiza a través de la intranet. Se ha configurado el *firewall* de ambas máquinas para permitir el tráfico desde internet solo al puerto de la API y de *adminer* y el tráfico *ssh* para su administración (permitiendo su acceso a través de un par de claves). El resto de tráfico es descartado. En el caso de la intranet solo está permitido el tráfico de comunicación con la base de datos entre las dos máquinas. Ambas máquinas tienen configurado un nombre de dominio ya que la IP que le asigna *Azure* a las máquinas es dinámica.

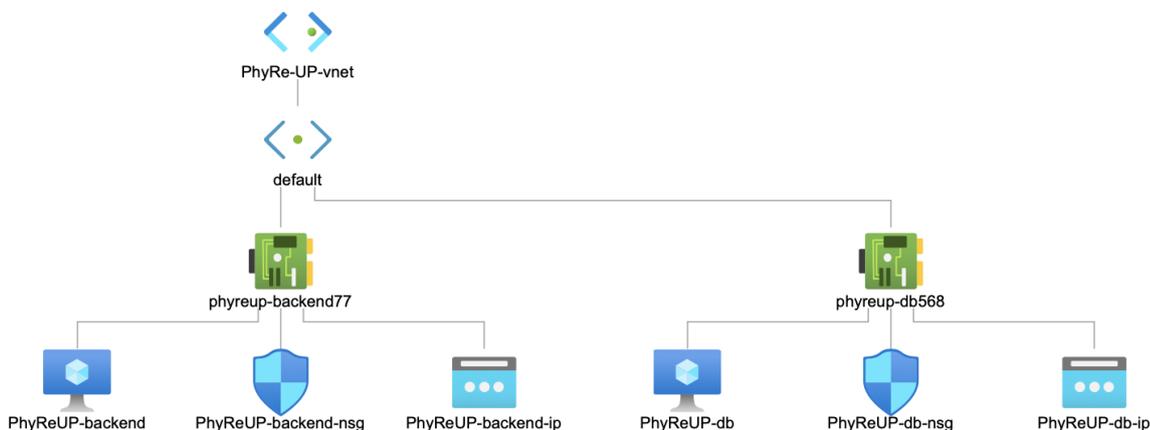


Figura 6.17: Topología del *backend* desde la vista de *Microsoft Azure*

6.2. DISTRIBUCIÓN DEL TRABAJO

En esta sección se van a explicar las iteraciones realizadas durante el desarrollo de este proyecto siguiendo la metodología explicada anteriormente, así como las divisiones en tareas y la planificación de las mismas. El sistema *PhyRe UP!* se va a dividir en cuatro paquetes de trabajo:

- Paquete 0. Documentación.
- Paquete 1. Aplicación de *tracking*.
- Paquete 2. Aplicaciones de RA para el terapeuta y paciente.
- Paquete 3. *Backend*.

6.2.1. Paquetes de trabajo

Documentación

Es el paquete de trabajo que engloba todas las tareas de documentación y estudio necesarias para el desarrollo del sistema *PhyRe UP!*. Se compone de cuatro tareas:

- Tarea 1: definición de requisitos del sistema. Es la tarea encargada de especificar los requisitos del sistema tras la reunión con los tutores del trabajo.
- Tarea 2: estudio de antecedentes de sistemas de rehabilitación de RA o RV. Esta tarea tiene como finalidad hacer un estudio de sistemas de rehabilitación ya desarrollados que usen técnicas de RA o RV.
- Tarea 3: estudio de modos de interacción de RA y RV. Para obtener la mejor experiencia para los usuarios se ha realizado un estudio sobre los diferentes modos de interacción y *feedback* en sistemas de RA y RV.
- Tarea 4: estudio de dispositivos hardware para elegir la opción más adecuada al proyecto. Una vez obtenidos elegidos el modo de interacción y que tipo de *feedback* va a recibir el usuario, se realiza una búsqueda y estudio de dispositivos *hardware* para encontrar el mejor para este sistema.

Aplicación de *tracking*

Este paquete de trabajo corresponde al desarrollo de la aplicación de *tracking* que hace uso del dispositivo *Azure Kinect DK*, que tiene como función detectar la posición de las manos del usuario y enviarlo a las aplicaciones de RA. Consta de tres tareas:

- Tarea 1: diseño e implementación del *tracking* y envío a la aplicación de RA. Esta tarea ha tenido como finalidad el diseño y realización de la aplicación que obtenga la posición de las manos y el envío a la aplicación de RA.
- Tarea 2: optimizar la obtención y envío de datos. A lo largo de esta tarea se ha realizado la optimización de la aplicación de *tracking* con el fin de obtener los mejores resultados.
- Tarea 3: validación y pruebas. Durante esta tarea se han realizado pruebas para comprobar que el funcionamiento de la aplicación es el correcto.

Aplicaciones de RA para el terapeuta y paciente

Este paquete de trabajo engloba las tareas de diseño y desarrollo de las aplicaciones de RA que usarán el paciente y el terapeuta. Se ha decidido no tener un paquete de trabajo para cada una de las aplicaciones puesto que ambas aplicaciones comparten varios módulos que solo deberán ser diseñados e implementados una vez.

- Tarea 1: desarrollo del módulo de detección de marcas. Durante esta tarea se ha buscado la mejor opción para la detección de marcas y se ha desarrollado el módulo encargado de esta función.
- Tarea 2: obtención y representación de datos de *tracking*. A lo largo de esta tarea se ha implementado la recepción de información de la aplicación de *tracking* y la representación de esta información al usuario.
- Tarea 3: diseño e implementación de la definición y asignación de juegos. Durante

esta tarea se ha llevado a cabo el diseño de los juegos y su representación. Además del diseño el diseño e implementación del módulo encargado de la definición de los ejercicios por parte del terapeuta y la asignación de estos juegos al paciente.

- Tarea 4: diseño del módulo de comunicación con el *backend* para la persistencia. Esta tarea sirvió para implementar el módulo de comunicación para dotar al sistema de persistencia mediante las llamadas al *backend*.
- Tarea 5: diseño e implementación de la realización de ejercicios. Durante esta tarea se diseñó e implementó el algoritmo de realización de ejercicios.
- Tarea 6: añadir técnicas de gamificación. A lo largo de esta tarea se añadieron técnicas de gamificación con la finalidad de aumentar la motivación del paciente mientras realice los ejercicios.
- Tarea 7: Validación y pruebas. A lo largo de esta tarea se han realizado pruebas para las dos aplicaciones RA con el fin de encontrar y solucionar los problemas que habían surgido durante el desarrollo.

Backend

Este paquete de trabajo tiene la responsabilidad del diseño e implementación del backend usado para dotar de persistencia al sistema, así como realizar la evaluación del progreso del paciente.

- Tarea 1: diseño e implementación de la base de datos. Durante esta tarea se diseñó e implementó la base de datos del sistema.
- Tarea 2: diseño e implementación de la API. A lo largo de esta tarea se ha diseñado e implementado la API Rest.
- Tarea 3: diseño e implementación de la evaluación de ejercicios. Tarea que ha tenido como finalidad el desarrollo de la evaluación de los resultados de un paciente.
- Tarea 4: validación y pruebas. Durante esta tarea se han realizado pruebas para comprobar que el funcionamiento era el deseado y así detectar y corregir errores de implementación.

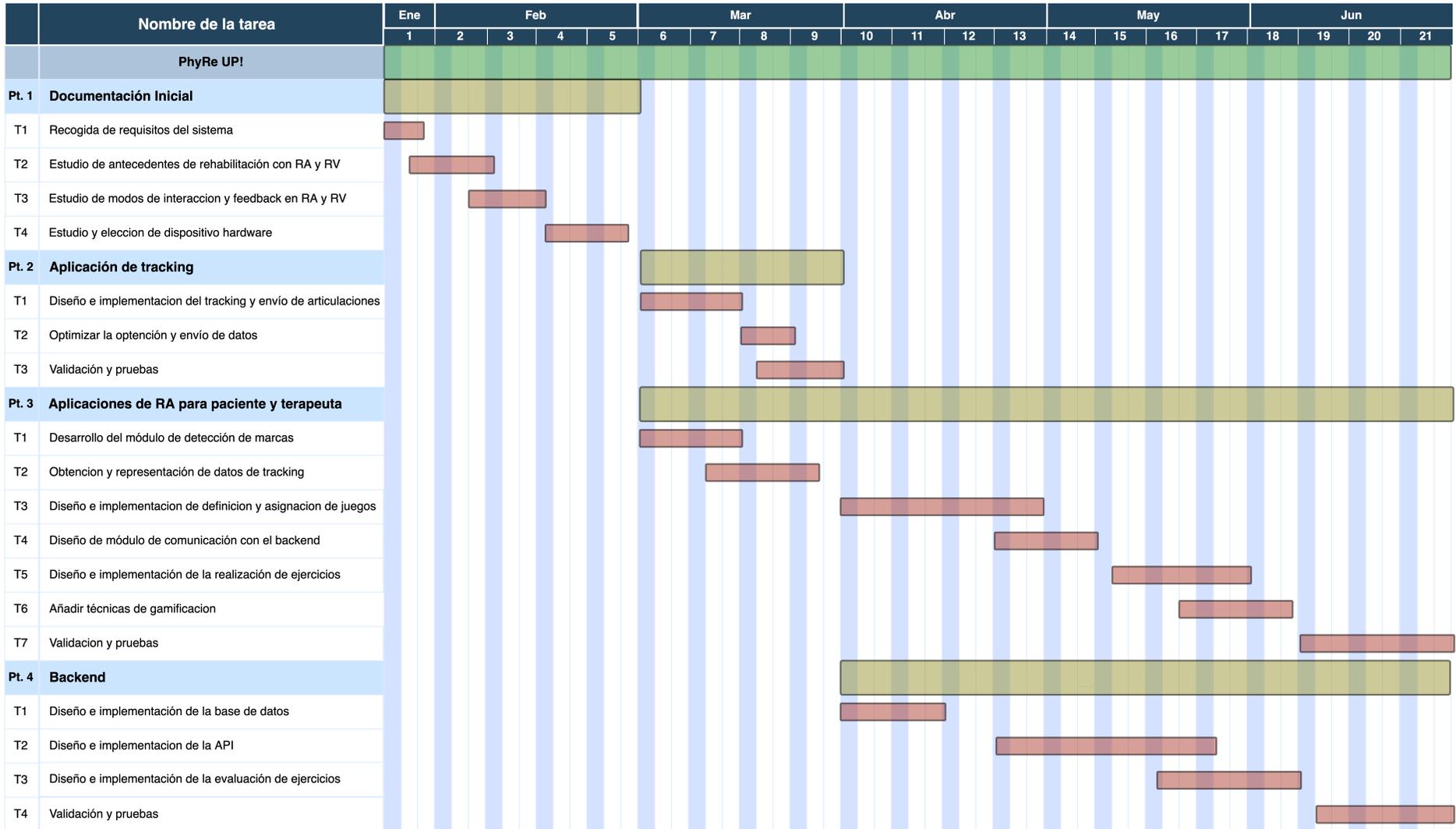


Figura 6.18: Diagrama de Gantt que representa la planificación del proyecto

6.2.2. Iteraciones

Este proyecto está compuesto de siete iteraciones. La primera iteración se centra en la especificación de requisitos, así como en el estudio de antecedentes y elección del *hardware* que ha sido usado en el proyecto. Las siguientes iteraciones se centran en el diseño e implementación del *software* de las cuatro aplicaciones que componen el sistema, y la última iteración en la validación y pruebas además del desarrollo de la memoria. En las tablas de resumen de iteración se puede observar el inicio y final de la iteración, los paquetes de trabajo implicados junto a sus tareas, una estimación de la duración y los objetivos alcanzados.

Iteración 1

La iteración 1 es la iteración de mayor duración. Esta iteración se ha centrado en la recogida de requisitos del sistema y en el estudio de sistemas de rehabilitación física que hagan uso de tecnologías de RA y RV, así como el estudio de los modos de interacción y retroalimentación de RA y RV. Para finalizar la iteración se ha hecho un estudio para encontrar el mejor dispositivo para el contexto de la rehabilitación. Ha sido una iteración clave puesto que estas investigaciones han servido de sustento para las siguientes iteraciones.

Iteración 1				
Inicio	27-enero-2020	Final:	28-febrero-2020	Estimación : 70h
Paquete de trabajo		Tareas	Objetivos alcanzados	
Pt. Documentación	T1	- Requisitos del sistema		
	T2	- Estudio de antecedentes		
	T3	- Dispositivos elegidos		
	T4	- Esquema de modos de interacción		

Tabla 6.1: Resumen de la iteración 1

Iteración 2

A lo largo de esta iteración y tomando como entrada el estudio sobre los diferentes modos de interacción y la elección de los dispositivos *hardware* más adecuados para este proyecto, se realizó el diseño y desarrollo de la aplicación de *tracking* y se empezó a diseñar e implementar la detección de marcas y la recepción de los datos de *tracking*. Como resultado se obtuvo una aplicación de prueba del sistema de tracking.

Iteración 2				
Inicio	02-marzo-2020	Final:	13-marzo-2020	Estimación : 40h
Paquete de trabajo		Tareas	Objetivos alcanzados	
Pt.1 Aplicación de tracking	T1	- Tracking de las manos del paciente		
Pt.2 Aplicación de RA	T1	- Recepción de tracking en App RA		
	T2	- Representación del tracking en RA		

Tabla 6.2: Resumen de la iteración 2

Iteración 3

Durante esta iteración se optimizó la aplicación de *tracking* para obtener el mayor rendimiento posible y se terminó la recepción y representación de los datos de *tracking* en la aplicación de RA.

Iteración 3				
Inicio	16-marzo-2020	Final:	27-marzo-2020	Estimación : 35h
Paquete de trabajo		Tareas	Objetivos alcanzados	
Pt.1 Aplicación de tracking		T2 T3	- Optimización de tracking - Validación y pruebas	
Pt.2 Aplicación de RA		T2	- Corrección de errores	

Tabla 6.3: Resumen de la iteración 3

Iteración 4

En esta iteración se diseñó la forma de los ejercicios y su representación, así como el algoritmo y los pasos que el terapeuta tendrá que hacer para definir los ejercicios y asignarlos al paciente. Además se diseñó e implementó la base de datos del sistema.

Iteración 4				
Inicio	30-marzo-2020	Final:	10-abril-2020	Estimación : 40h
Paquete de trabajo		Tareas	Objetivos alcanzados	
Pt.2 Aplicación de RA		T3	- Diseño de juegos - Representación de juegos	
Pt.3 Backend		T1	- Diseño e implementación de la base de datos	

Tabla 6.4: Resumen de la iteración 4

Iteración 5

A lo largo de esta iteración se desarrolló la API para dotar de persistencia al sistema. Así como el módulo de comunicación para que las aplicaciones de RA puedan comunicarse con la API y se terminó la definición y asignación de ejercicios.

Iteración 5				
Inicio	13-abril-2020	Final:	04-mayo-2020	Estimación : 55h
Paquete de trabajo		Tareas	Objetivos alcanzados	
Pt.2 Aplicación de RA		T3 T4	- Asignación de juegos - Comunicación con API - Guardar juegos y asignaciones en el backend	
Pt.3 Backend		T2	- API implementada parcialmente	

Tabla 6.5: Resumen de la iteración 5

iteración 6

A lo largo de esta iteración se terminó la implementación de la API y se diseñó el algoritmo para la comparación de ejercicios y la evaluación de estos ejercicios. Además del diseño del *feedback* que recibe el paciente y se añadieron las técnicas de gamificación en la realización de los ejercicios.

Iteración 6					
Inicio	11-mayo-2020	Final:	29-mayo-2020	Estimación	: 60h
Paquete de trabajo	Tareas	Objetivos alcanzados			
Pt.2 Aplicación de RA	T5 T6	- Comparación de ejercicios - Gamificación			
Pt.3 Backend	T2 T3	- API implementada - Evaluación de resultados			

Tabla 6.6: Resumen de la iteración 6

Iteración 7

Esta es la iteración final, en la cual se llevaron a cabo las tareas de validación y pruebas para las aplicaciones de RA y se corrigieron los errores encontrados. Por último, se preparó el documento final que contiene la memoria del proyecto.

Iteración 7					
Inicio	01-junio-2020	Final:	22-junio-2020	Estimación	: 90h
Paquete de trabajo	Tareas	Objetivos alcanzados			
Pt.0 Documentación		- Memoria			
Pt.2 Aplicación de RA	T7	- Validación y pruebas - Corrección de errores			
Pt.3 Backend	T4	- Validación y pruebas - Corrección de errores - Despliegue en la nube			

Tabla 6.7: Resumen de la iteración 7

6.3. COSTES DE DESARROLLO

El desarrollo del sistema *PhyRe UP!* se inicia el 27 de enero de 2020 y finaliza el 21 de junio de 2020, es decir, 21 semanas de trabajo. Se han dedicado 390 horas en el desarrollo del proyecto. Según la página *indeed*¹, el sueldo medio de un desarrollador *software* en España está en torno a los 17.58€/h bruto. Contando con que para este proyecto solo ha habido un desarrollador el coste se sitúa en 6.856,2€.

Como se ha comentado anteriormente, el *backend* está desplegado en dos máquinas de *Microsoft Azure*, cuya facturación se realiza por el tiempo de uso de las máquinas (tiempo

¹<https://es.indeed.com/salaries/desarrollador-de-software-Salaries?period=hourly>

que las máquinas estén encendidas), siendo un precio de 0,0398 €/h por máquina. Contado con que las máquinas estén 24/7 encendidas se deduce un gasto aproximado de 29,06 €/mes por cada máquina. Puesto que el despliegue se ha realizado en la última iteración solo se va a contar el coste de las máquinas de un solo mes.

Para el desarrollo del proyecto se han usado dos ordenadores, un *MacBook Pro* con un coste de 2200€ y un *Asus R510V* con un coste de 650€. Además, para el proyecto se han necesitado de dos dispositivos: el dispositivo de realidad aumentada *Microsoft HoloLens 1* con un coste de 3000€ y el dispositivo *Azure Kinect DK* usado para el *tracking* con un coste de 354€.

Por tanto, de los datos anteriores resulta un coste total del proyecto de 13.118,32 €. En la [Tabla 6.8](#) se puede observar un resumen de los costes del proyecto.

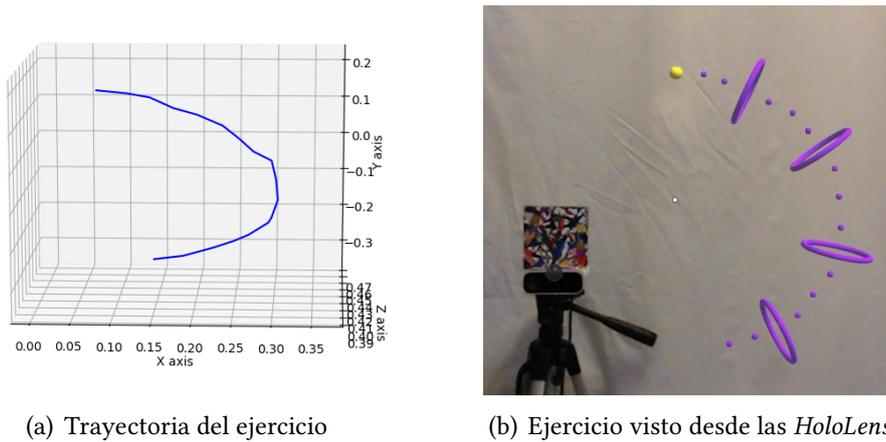
	Recurso	Unidades	Precio/unidad	Precio
Recursos para el desarrollo	MacBook Pro	1 u.	2.200 €/u	2.200 €
	Asus R510V	1 u.	650 €/u	650 €
Recursos del proyecto	HoloLens	1 u.	3.000 €/u	3.000 €
	Azure Kinect	1 u.	354 €/u	354 €
	Máquina B2s estándar	2 u.	29,06 €/u	58,12 €
Recursos humanos	Desarrollador	500 h	17,58 €/h	6.856,2 €
Coste Total				13.118,32 €

Tabla 6.8: Coste total del proyecto

6.4. EXPERIMENTACIÓN

En esta sección se van a mostrar las pruebas realizadas con el sistema en diferentes sujetos. Dada la extraordinaria situación que acontece el año 2020 debido a la pandemia del *SARS-CoV-2*, solo se han podido realizar pruebas del sistema con personas del círculo cercano del autor y no con personas con necesidad de rehabilitación para las que está pensado este sistema.

Las pruebas consisten en mostrar a los sujetos un ejercicio que deben hacer y lo realizarán con y sin el sistema, para comprobar si haciendo uso de la aplicación realizarán el ejercicio con más exactitud. Para realizar esta tarea se ha implementado una aplicación para las *HoloLens* sin ningún tipo de *feedback* que se encargará de registrar la posición de la mano del paciente enviada por la aplicación de *tracking*, también se ha modificado la aplicación del paciente para que guarde las posiciones de la articulación durante la realización del ejercicio y así poder compararlas. En el [Anexo-B](#) se muestra el *script* usado para la representación y un ejemplo de los datos recogidos de la articulación del paciente.

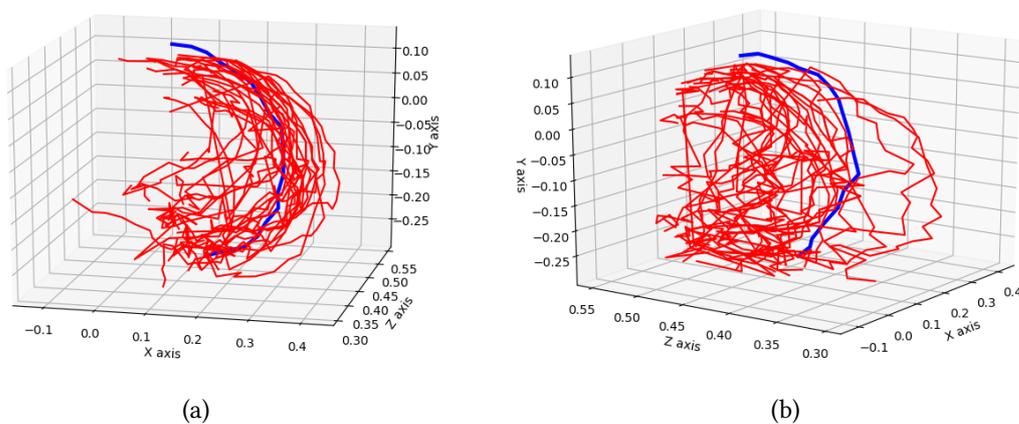


(a) Trayectoria del ejercicio

(b) Ejercicio visto desde las HoloLens

Figura 6.19: Ejercicio definido para las pruebas

El autor ha definido un ejercicio y se ha grabado realizándolo. Este vídeo realizando el ejercicio se les ha mostrado a los sujetos que participan en la experimentación, y se les han proporcionado las instrucciones oportunas para la realización del ejercicio. El ejercicio que se ha definido se puede observar en la Figura 6.19, en esta figura se observa la trayectoria tridimensional del ejercicio y la vista que verán los pacientes cuando usen la aplicación de RA.



(a)

(b)

Figura 6.20: Resultado del ejercicio sin *feedback*. En azul el ejercicio definido por el terapeuta y en rojo las trayectorias trazadas por los sujetos de prueba

En total cada sujeto del experimento ha realizado cinco repeticiones del ejercicio sin *feedback* para, posteriormente realizarlo haciendo uso de la aplicación desarrollada. En la Figura 6.20) se puede observar los resultados tras la realización del ejercicio sin *feedback*. La línea azul es el ejercicio definido y marcado por el terapeuta y las líneas rojas son las trayectorias trazadas por los sujetos al realizar el ejercicio. Como se puede observar, las trayectorias realizadas por los sujetos difieren mucho de la trayectoria del ejercicio. El eje donde más discrepa con el ejercicio definido por el terapeuta es el eje «Z», esto puede ser

debido a que al mostrar el ejercicio en un vídeo, la profundidad es lo más difícil de visionar y recrear.

Tras esta prueba donde los sujetos han realizado el ejercicio sin *feedback*, ha realizado el mismo ejercicio con cinco repeticiones, pero haciendo uso de la aplicación desarrollada. En la Figura 6.21 se pueden observar los resultados. En este caso, se observa como las trayectorias realizadas por los sujetos se asemejan más a la definida por el terapeuta. A pesar de la simplicidad del ejercicio usado para este experimento, se observan grandes diferencias en la precisión entre ambos modos. Esto da a entender que con ejercicios más complejos las diferencias serian aún mayores.

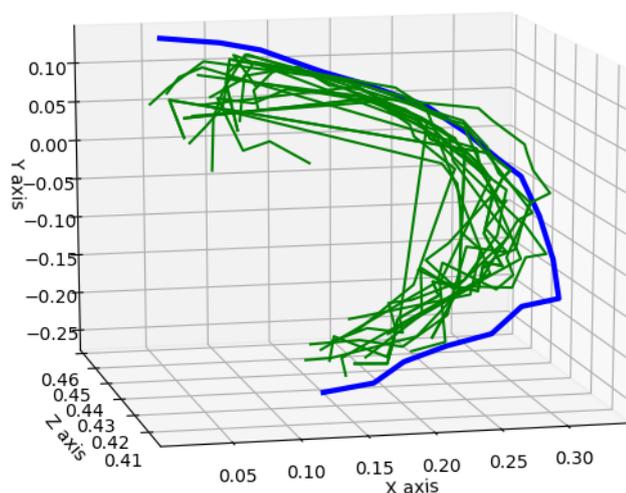
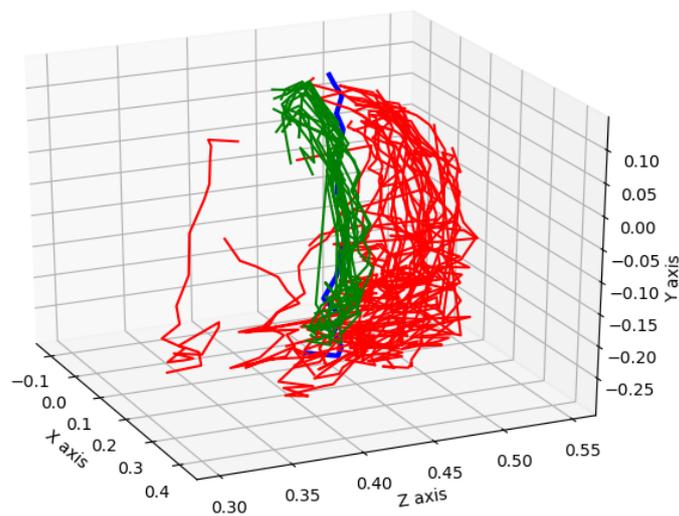


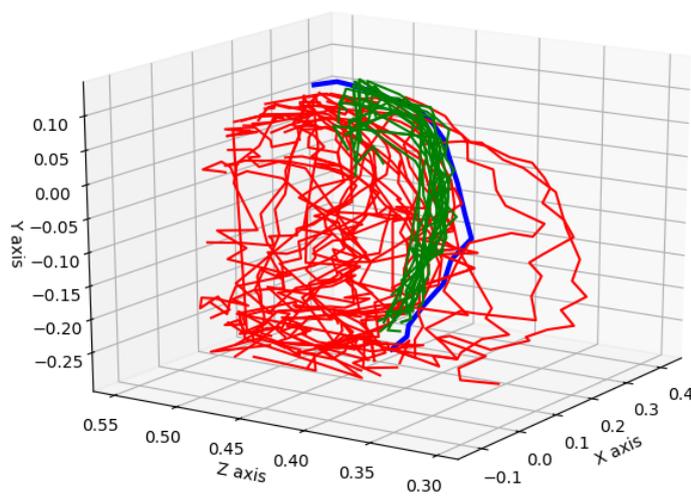
Figura 6.21: Trayectorias trazadas por los sujetos del experimento haciendo uso de la aplicación de RA. En azul el ejercicio definido por el terapeuta y en verde las trayectorias trazadas por los sujetos de prueba

En la Figura 6.22 se puede comparar las trayectorias con *feedback* (verde) y sin *feedback* (rojo). En estas figuras se observa como haciendo uso de la aplicación de RA, los sujetos realizan un ejercicio más acorde al definido por el terapeuta. En estas figuras se puede apreciar una diferencia significativa en la recreación de los ejercicios entre pacientes que no usan el sistema desarrollado en este TFG y los que sí lo usan. En verde, se representan ejercicios realizados por pacientes que reproducen un ejercicio definido por un terapeuta (en azul), mientras lo visualizan con técnicas de RA. Tal como se puede apreciar, el mismo movimiento realizado por el terapeuta se recrea con gran precisión por parte del paciente. Sin embargo, en rojo, los movimientos realizados por pacientes sin el uso del sistema propuesto no se corresponden fielmente con lo definido por el terapeuta, a pesar de que el ejercicio propuesto es simple y fácil de recordar, lo que significa que ante ejercicios más complejos la diferencia sería aún mayor.

Con ello, se pretende demostrar que el sistema propuesto, además de buscar la motivación del paciente y su constancia, también persigue mejorar la calidad del proceso de rehabilitación autónomo sin presencia del terapeuta.



(a)



(b)

Figura 6.22: Comparación del resultado con y sin *feedback*. En azul el ejercicio definido por el terapeuta, en rojo las trayectorias trazadas por los sujetos de prueba sin *feedback* y en verde con *feedback*

CAPÍTULO 7

CONCLUSIONES

En este capítulo se realizará un juicio crítico y discusión sobre los resultados obtenidos. En primer lugar, se realizará un pequeño resumen del contexto, cómo este trabajo pretende darle solución a la problemática y la opinión del autor. Después se revisarán los objetivos iniciales del trabajo para discutir si han sido alcanzados tras el desarrollo del proyecto. Seguidamente se hará una justificación de las competencias adquiridas de la tecnología específica de *ingeniería de computadores* para concluir con las posibles líneas de trabajo futuro.

La rehabilitación física de pacientes supone reunir a terapeuta y paciente en el mismo lugar físico. Durante esta estancia el paciente realiza los ejercicios mandados por el terapeuta mientras este lo guía. Puesto que el tiempo que el terapeuta puede pasar con el paciente es limitado, los ejercicios que el paciente no ha podido realizar en la consulta deberá realizarlos en su domicilio, el terapeuta deberá asumir que el paciente está realizando los ejercicios correctamente y hasta la próxima consulta no podrá conocer su avance. Otro problema de la rehabilitación física tradicional reside en la motivación de los pacientes, una pérdida de motivación puede ocasionar que el paciente abandone los ejercicios.

Como solución a estas problemáticas expuestas se presenta el sistema *PhyRe UP!*. Este sistema hace uso de la realidad aumentada para que el paciente pueda realizar los ejercicios de rehabilitación de forma autónoma y el terapeuta pueda llevar un seguimiento del paciente. En este contexto, el terapeuta define y asigna los ejercicios al paciente y éste los realiza de forma guiada en su domicilio a través del sistema de RA. Los ejercicios se muestran como juegos al paciente y al hacer uso de técnicas de gamificación, se pretende que la motivación no disminuya y, por consiguiente, se evite el abandono de los ejercicios.

Para el desarrollo del sistema *PhyRe UP!*, se han desarrollado cuatro aplicaciones. Una aplicación para el *tracking* del cuerpo del paciente/terapeuta que hace uso del dispositivo *Azure Kinect DK* y envía esta información a la aplicación de RA que se esté usando. Se han desarrollado dos aplicaciones de RA para el dispositivo *HoloLens*, una para el terapeuta con la que podrá definir y asignar los ejercicios para los pacientes, y otra aplicación para

los pacientes con la que realizarán dichos ejercicios. Para finalizar, se ha implementado un *backend* con una *API Rest* y una base de datos, que incluye lógica además de dotar de persistencia al sistema.

Tal como se muestra en el Capítulo-6, la experimentación con pacientes de prueba ha demostrado que el sistema ayuda de forma notable a los pacientes a realizar el ejercicio con calidad y exactitud a como lo ha definido el terapeuta. Además, de buscar su motivación y constancia al presentar los ejercicios en forma de juegos. En este capítulo se puede observar también los resultados de las aplicaciones desarrolladas.

A lo largo de este proyecto he aplicado gran parte de los conocimientos adquiridos a lo largo del grado para la construcción de las distintas partes del sistema. Siendo de especial mención los conocimientos aplicados de mi intensificación Ingeniería de Computadores que están explicados en la Sección-7.2. Además de aplicar los conocimientos técnicos desarrollados durante el grado, también he aplicado habilidades no técnicas trabajadas en el grado que me han ayudado en el auto-aprendizaje e investigación necesarias para desarrollar *PhyRe UP!*, por lo que para el desarrollo de este proyecto he adquirido nuevos conocimientos entre los que pueden destacar *unity*, realidad aumentada y el uso de servicios *cloud*.

7.1. OBJETIVOS ALCANZADOS

En esta sección se van a revisar los objetivos detallados en el Capítulo-2 con la idea de comprobar si se han alcanzado tras el desarrollo del proyecto.

1. *Estudio y evaluación de dispositivos hardware*. Al inicio del proyecto, tras la especificación de requisitos, se realizó un análisis de diferentes dispositivos actuales de RA y *tracking* de cuerpo, para concluir que la mejor elección para el desarrollo de un sistema de rehabilitación cómodo de usar y que otorgue un buen *feedback* al usuario, es la combinación de los dispositivos punteros *Azure Kinect DK* para el *tracking* y *Microsoft HoloLens 1* como dispositivo de RA.
2. *Diseño y desarrollo de esquemas de interacción basados en realidad aumentada atendiendo a las capacidades computacionales del dispositivo escogido para el proyecto*. Gracias a la capacidad de cómputo del dispositivo de RA elegido y del uso de un sistema de *tracking* sin marcas se ha conseguido que los esquemas de interacción con el entorno de RA no supongan una molestia desde la vista del paciente. La interacción a través de gestos, mirada y voz hacen que el paciente no tenga que usar controles para manejar la aplicación. Además, no es necesario el uso de marcas físicas en el cuerpo del paciente para poder realizar el *tracking* de las extremidades.
3. *Desarrollo de un módulo software que permita al terapeuta definir los ejercicios*. Para

satisfacer este objetivo se ha diseñado e implementado una aplicación de RA con la arquitectura mostrada en el capítulo-5 que permite al terapeuta definir los ejercicios y asignarlos al paciente. Esta aplicación cuenta con una interfaz de usuario amigable.

4. *Desarrollo de un backend que permita almacenar los ejercicios y los resultados de los mismos.* Dicho objetivo ha quedado cumplido con la elaboración de un *backend* con una *API Rest* y una base de datos para dotar de persistencia al sistema. Además, se ha realizado un despliegue en la nube de *Azure* para que tanto el paciente como el terapeuta puedan acceder desde cualquier lugar.
5. *Desarrollo de un módulo software basado en gamificación orientado a motivar a los pacientes para la realización de los ejercicios.* A través de la aplicación de RA desarrollada, el paciente puede realizar los ejercicios que consisten en unas esferas (puntos) y aros que deberá ir alcanzando. El paciente recibirá un *feedback* visual y sonoro cuando obtenga los puntos, además de guiar al paciente si éste se está equivocando. Cuando alcance todos los puntos se terminará una repetición del ejercicio. De esta manera, al paciente se le presenta el ejercicio como un juego aumentando así su motivación y reduciendo la probabilidad de abandono.
6. *Desarrollo de un módulo software que permita evaluar automáticamente los ejercicios de rehabilitación realizados por un paciente.* Este objetivo se ha alcanzado con el desarrollo de dos módulos. Uno de ellos en la aplicación del paciente encargado de comparar la posición de la articulación con los puntos clave del ejercicio para la realización de los ejercicios, y otro módulo en el *backend* encargado de evaluar el progreso del paciente con sus registros anteriores.

7.2. JUSTIFICACIÓN DE COMPETENCIAS ADQUIRIDAS

El desarrollo del presente TFG ha servido para poner en práctica gran parte de los conocimientos adquiridos durante el Grado en Ingeniería Informática, y adquirir otros nuevos que han contribuido en la adquisición de competencias específicas de la intensificación de *Ingeniería de Computadores*.

IC4: *Capacidad de diseñar e implementar software de sistema y de comunicaciones.* Para la elaboración de este TFG se ha necesitado la construcción de un sistema de comunicación para el envío de datos entre la aplicación de *tracking* del cuerpo y la aplicación de RA, así como la comunicación de las aplicaciones de RA con la *API Rest*. Además, ha sido necesaria el desarrollo del *software* para las aplicaciones de RA y de *tracking*

del cuerpo.

IC7: *Capacidad para analizar, evaluar, seleccionar y configurar plataformas hardware para el desarrollo y ejecución de aplicaciones y servicios informáticos.* Para el desarrollo de este proyecto se ha tenido que evaluar diferentes dispositivos *hardware* de RA y de *tracking* de cuerpo para elegir los mejores dispositivos para la propuesta que aquí se expone. Además, se ha adaptado los modos de interacción con el sistema atendiendo a las capacidades computacionales de los dispositivos.

IC8: *Capacidad para diseñar, desplegar, administrar y gestionar redes de computadores.* Para la elaboración de este trabajo se ha diseñado una red de computadores que alberga el *backend* del sistema, configurando las interfaces, direcciones y el *firewall* de los equipos además de su despliegue y configuración en la plataforma de computación en la nube de *Microsoft Azure*.

7.3. TRABAJO FUTURO

Para finalizar con este TFG se van a explicar las líneas de trabajo futuro más destacable para este proyecto:

- A finales de 2020 se comercializará el nuevo dispositivo *Microsoft HoloLens 2*, que cuenta con una mayor capacidad de computo, así como cámaras y sensores de última generación. Se podría explorar la posibilidad de migrar el sistema para hacer uso de este nuevo dispositivo que cuenta con mejores especificaciones, lo que podría dotar al sistema de una mayor fluidez y versatilidad.
- Permitir al paciente emitir el ejercicio que está realizando en ese momento para que así, el terapeuta, mediante el uso de otro dispositivo de RA pudiera conectarse a observar con detalle el ejercicio que está realizando el paciente en tiempo real.
- Para este trabajo solo se ha diseñado un tipo de juego que consiste en seguir trayectorias que el terapeuta define. Para nuevas versiones se pueden incluir diferentes tipos de juego con el fin de reducir la monotonía y así evitar la desmotivación del paciente.

BIBLIOGRAFÍA

- [1] J. Alvarez-Sabín y col., «Economic impact of patients admitted to stroke units in Spain», *The European Journal of Health Economics*, vol. 18, n.º 4, págs. 449-458, 2017.
- [2] B. Aruanno, F. Garzotto y M. C. Rodriguez, «HoloLens-based Mixed Reality Experiences for Subjects with Alzheimer's Disease», en *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter*, 2017, págs. 1-9.
- [3] Y. M. Aung y A. Al-Jumaily, «Augmented reality-based RehaBio system for shoulder rehabilitation», *International Journal of Mechatronics and Automation*, vol. 4, n.º 1, págs. 52-62, 2014.
- [4] P. A. Bernstein y N. Goodman, «Multiversion concurrency control—theory and algorithms», *ACM Transactions on Database Systems (TODS)*, vol. 8, n.º 4, págs. 465-483, 1983.
- [5] J. E. Bos, W. Bles y E. L. Groen, «A theory on visually induced motion sickness», *Displays*, vol. 29, n.º 2, págs. 47-57, 2008.
- [6] A. J. i Capó y col., «Motivational rehabilitation using serious games», *Virtual Archaeology Review*, vol. 4, n.º 9, págs. 167-173, 2013.
- [7] G. T. del Castillo, «La representación de rotaciones mediante cuaterniones», *Miscelanea Matemtica*, págs. 43-50, 1999.
- [8] M. A. Cidota, P. J. Bank y S. G. Lukosch, «Design Recommendations for Augmented Reality Games for Objective Assessment of Upper Extremity Motor Dysfunction», en *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, págs. 1430-1438.
- [9] A. E. F. Da Gama y col., «MirrARbilitation: A clinically-related gesture recognition interactive tool for an AR rehabilitation system», *Computer methods and programs in biomedicine*, vol. 135, págs. 105-114, 2016.
- [10] F. Debandi y col., «Enhancing cultural tourism by a mixed reality application for outdoor navigation and information browsing using immersive devices», en *IOP Conference Series: Materials Science and Engineering*, vol. 364, 2018, pág. 012 048.
- [11] S. Deterding y col., «From game design elements to gamefulness: defining "gamification"», en *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, 2011, págs. 9-15.

- [12] P. Epure y M. B. Holte, «Analysis of Motivation in Virtual Reality Stroke Rehabilitation», en *6th International Conference, ArtsIT 2017, and Second International Conference, DLI 2017*, Springer, 2018, págs. 282-293.
- [13] R. T. Fielding, «REST: architectural styles and the design of network-based software architectures», *Doctoral dissertation, University of California*, 2000.
- [14] E. Geschwinde y H.-J. Schönig, *PostgreSQL developer's handbook*. Sams Publishing, 2002.
- [15] M. D. Griffiths, «The educational benefits of videogames», *Education and health*, vol. 20, n.º 3, págs. 47-51, 2002.
- [16] P. G. D. Group. (2020). Postgresql license, PostgreSQL Global Development Group, [Online]. Available: <https://www.postgresql.org/about/licence/> (visited on 04/10/2020).
- [17] J. J. LaViola Jr, «A discussion of cybersickness in virtual environments», *ACM Sigchi Bulletin*, vol. 32, n.º 1, págs. 47-56, 2000.
- [18] B. D. Lawson y col., «Signs and symptoms of human syndromes associated with synthetic experiences.», 2002.
- [19] B. Loreto Vergara, «Desarrollo de la Medicina Física y Rehabilitación como especialidad médica», 2014.
- [20] N. Maclean y col., «A critical review of the concept of patient motivation in the literature on physical rehabilitation», *Soc Sci Med*, vol. 50, n.º 4, págs. 495-506, 2000.
- [21] D. Martin, «Big Data Visualization: HoloLens Brings Meaningful Interaction to Lupus Medical Data», *The Journal of Purdue Undergraduate Research*, vol. 8, n.º 1, pág. 32, 2018.
- [22] S. A. Maulden y col., «Timing of initiation of rehabilitation after stroke», *Archives of physical medicine and rehabilitation*, vol. 86, n.º 12, págs. 34-40, 2005.
- [23] S. McCallum, «Gamification and serious games for personalized health.», en *pHealth*, 2012, págs. 85-96.
- [24] D. Merkel, «Docker: lightweight linux containers for consistent development and deployment», *Linux journal*, vol. 2014, n.º 239, pág. 2, 2014.
- [25] Microsoft. (2020). Microsoft hololens, Microsoft, [Online]. Available: <https://www.microsoft.com/en-us/hololens> (visited on 04/10/2020).
- [26] c. y. b. s. Ministerio de sanidad, «Encuesta nacional de salud. España», 2018.
- [27] M. Montagnini, N. M. Javier y C. Ritchie, «Physical therapy and other rehabilitation issues in the palliative care setting», *UpToDate May*, vol. 16, 2017.
- [28] H. Mousavi Hondori y col., «Choice of human-computer interaction mode in stroke rehabilitation», *Neurorehabilitation and neural repair*, vol. 30, n.º 3, págs. 258-265, 2016.
- [29] J. Nickolls y col., «Scalable parallel programming with CUDA», *Queue*, vol. 6, n.º 2, págs. 40-53, 2008.

- [30] E. J. O’Neil, «Object/relational mapping 2008: hibernate and the entity data model (edm)», en *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, págs. 1351-1356.
- [31] J. Odom, *HoloLens Beginner’s Guide*. Packt Publishing Ltd, 2017.
- [32] J. Olesen y col., «The economic cost of brain disorders in Europe», *European journal of neurology*, vol. 19, n.º 1, págs. 155-162, 2012.
- [33] W. H. Organization y col., *Telemedicine: opportunities and developments in member states. Report on the second global survey on eHealth*. World Health Organization, 2010.
- [34] J. Piaget y B. Inhelder, *Psicología del niño*. Ediciones Morata, 1997, vol. 369.
- [35] P. Pratt y col., «Through the HoloLens™ looking glass: augmented reality for extremity reconstruction surgery using 3D vascular models with perforating vessels», *European radiology experimental*, vol. 2, n.º 1, pág. 2, 2018.
- [36] P. J. Rosa y col., «The effect of virtual reality-based serious games in cognitive interventions: a meta-analysis study», en *Proceedings of the 4th Workshop on ICTs for improving Patients Rehabilitation Research Techniques*, 2016, págs. 113-116.
- [37] L. F. dos Santos y col., «Visualization of virtual reality neurological motor rehabilitation of the upper limb—a systematic review», en *2013 International Conference on Virtual Rehabilitation (ICVR)*, IEEE, 2013, págs. 176-177.
- [38] B. Sawyer y P. Smith, «Serious games taxonomy», en *Slides from the Serious Games Summit at the Game Developers Conference*, vol. 5, 2008.
- [39] Y. A. Sekhavat y M. S. Namani, «Projection-based AR: Effective visual feedback in gait rehabilitation», *IEEE Transactions on Human-Machine Systems*, vol. 48, n.º 6, págs. 626-636, 2018.
- [40] A. Seyedebrahimi, R. Khosrowabadi y H. M. Hondori, «Brain Mechanism in the Human-Computer Interaction Modes Leading to Different Motor Performance», en *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, IEEE, 2019, págs. 1802-1806.
- [41] M. Shaughnessy, B. M. Resnick y R. F. Macko, «Testing a model of post-stroke exercise behavior», *Rehabilitation nursing*, vol. 31, n.º 1, págs. 15-21, 2006.
- [42] M. Sousa y col., «SleeveAR: Augmented reality for rehabilitation using realtime feedback», en *Proceedings of the 21st international conference on intelligent user interfaces*, 2016, págs. 175-185.
- [43] V. E. Team, «Vuforia developer portal», *Recuperado de: <https://developer.vuforia.com>*, 2020.
- [44] B. Terrill, «My coverage of lobby of the social gaming summit», *Bret on Social Games*, 2008.
- [45] *typeorm*, 2020. dirección: <https://github.com/typeorm/typeorm>.
- [46] L. Vera y col., «Augmented mirror: interactive augmented reality system based on kinect», en *IFIP Conference on Human-Computer Interaction*, Springer, 2011, págs. 483-486.

-
- [47] R. M. Viglialoro y col., «Review of the augmented reality systems for shoulder rehabilitation», *Information*, vol. 10, n.º 5, pág. 154, 2019.
- [48] G. Xylomenos y G. C. Polyzos, «TCP and UDP performance over a wireless LAN», en *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, IEEE, vol. 2, 1999, págs. 439-446.
- [49] Y. Zhang y col., «Application of Mixed Reality Based on Hololens in Nuclear Power Engineering», en *International Symposium on Software Reliability, Industrial Safety, Cyber Security and Physical Protection for Nuclear Power Plant*, Springer, 2019, págs. 9-20.
- [50] Z. Zhang, «Microsoft kinect sensor and its effect», *IEEE multimedia*, vol. 19, n.º 2, págs. 4-10, 2012.
- [51] M. Zyda, «From visual simulation to virtual reality to games», *Computer*, vol. 38, n.º 9, págs. 25-32, 2005.

ANEXOS

FOTOGRAFÍA DEL SISTEMA

En la Figura A.1 se puede observar el sistema el escenario físico y los dispositivos para la rehabilitación con *PhyRe UP!*. Se puede observar las *HoloLens* (1), el dispositivo *Azure Kinect DK* para el *tracking* del cuerpo del paciente (2), el *PC Host* (3) al que está conectado el dispositivo *Azure Kinect DK*, encargado de procesar las imágenes y enviar la información a la aplicación de RA. Por último, podemos observar la mano del paciente (4) mientras interactúa con la aplicación.

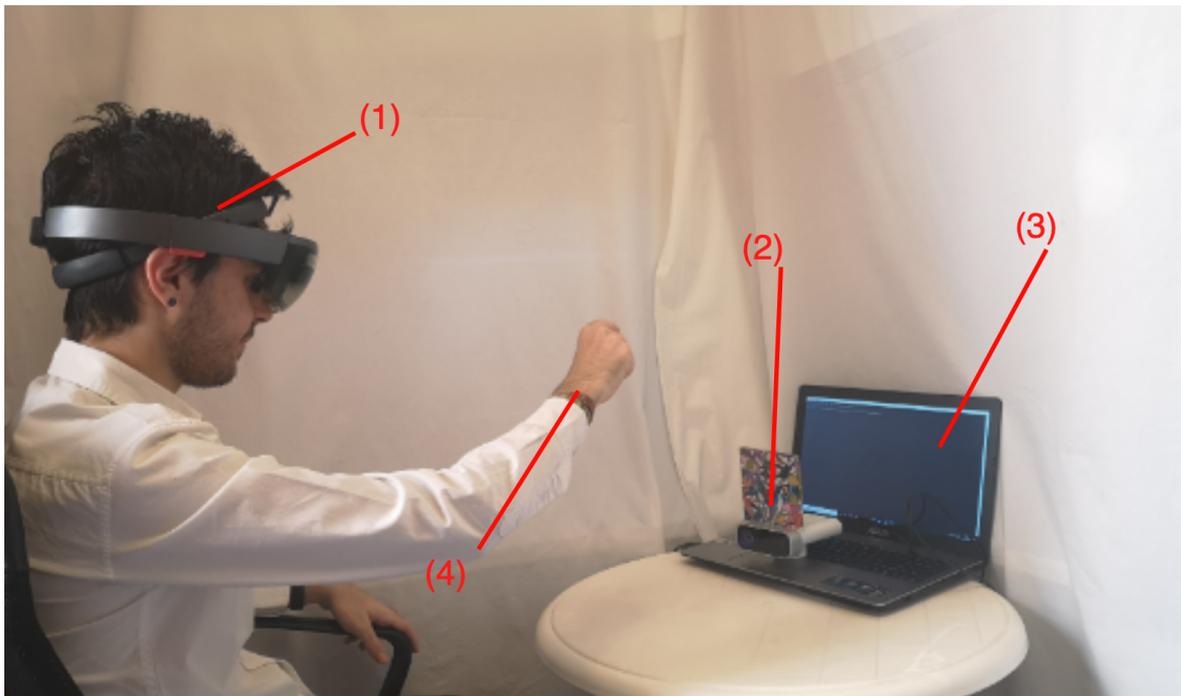


Figura A.1: Fotografía del sistema *PhyRe UP!* con anotaciones

En la figura A.2 se puede observar la fotografía con mejor calidad y sin anotaciones.



Figura A.2: Fotografía del sistema *PhyRe UP!* sin anotaciones

EXPERIMENTACIÓN

En este anexo se va a explicar cómo se han tomado los datos que se observan en las gráficas de la Sección-6.4 del capítulo de resultados. En primer lugar, el terapeuta ha definido un ejercicio a través de la aplicación de RA. El ejercicio que el terapeuta a definido se puede observar en la Figura B.1.

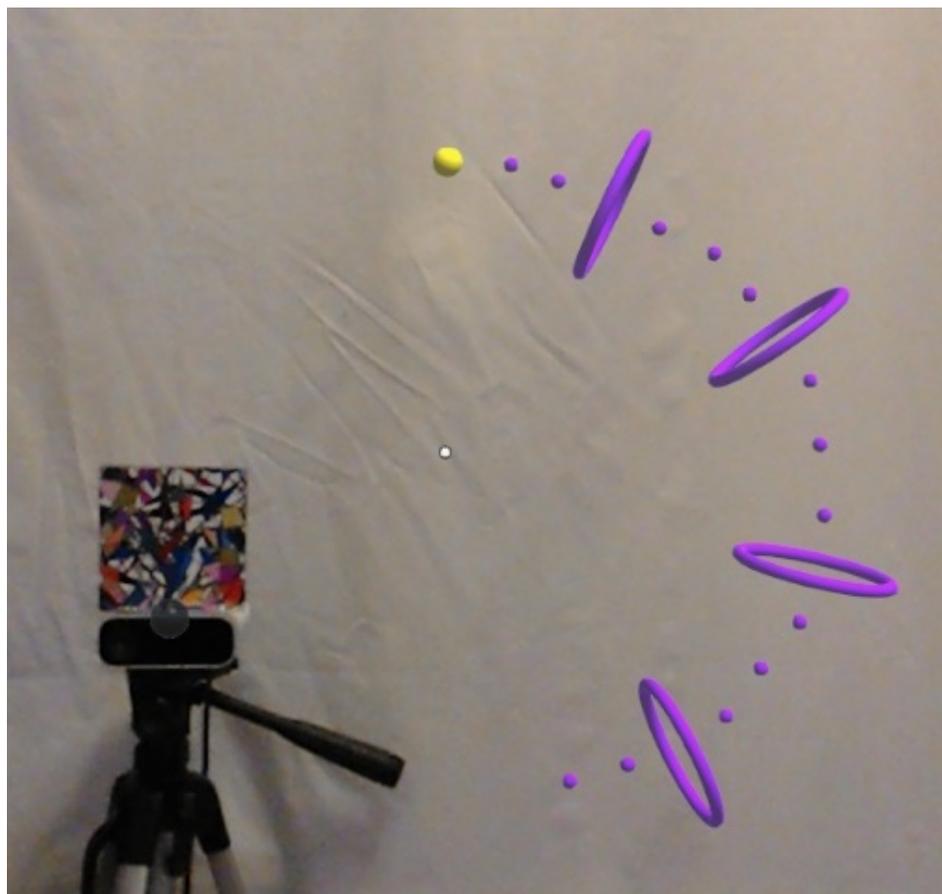


Figura B.1: Ejercicio visto desde la Aplicación de RA

Los pacientes harán uso de una aplicación para las *HoloLens* sin ningún *feedback* y de la aplicación con el sistema *PhyRe UP!* para realizar el ejercicio como se ha mostrado y explicado al paciente. Las aplicaciones registrarán la posición de la mano del paciente guardando los puntos por donde el paciente pasa. Con el fin de no generar un archivo demasiado extenso de los puntos de la trayectoria del paciente, solo se guardarán los puntos que tengan 1 cm de separación entre ellos. Un ejemplo de cómo se representará la trayectoria se puede observar en el Listado B.1.

Listado B.1: Ejemplo de archivo csv con la trayectoria del paciente

```

1
2 0.06028778851032257,0.13350822031497955,0.4584451913833618
3 0.10136336088180542,0.1323951631784439,0.4532321095466614
4 0.1325763463973999,0.12253408133983612,0.45256507396698
5 0.16655893623828888,0.107828788459301,0.44157373905181885
6 0.19747284054756165,0.09295202791690826,0.43994417786598206
7 0.23235034942626953,0.0768832117319107,0.4303101599216461
8 0.25475645065307617,0.0508168488740921,0.42415881156921387
9 0.27315741777420044,0.023340359330177307,0.4203373193740845
10 0.2960754632949829,-0.004646122455596924,0.42356565594673157
11 0.3023639917373657,-0.04602035880088806,0.41829898953437805
12 0.3052248954772949,-0.09168584644794464,0.41317611932754517
13 0.2978842258453369,-0.12616971135139465,0.40564149618148804
14 0.2927134931087494,-0.1624394655227661,0.4202284812927246
15 0.2677280306816101,-0.1928567886352539,0.41898632049560547
16 0.247731015086174,-0.22593778371810913,0.4290018677711487
17 0.21910333633422852,-0.25093188881874084,0.4324299395084381
18 0.18285390734672546,-0.25704896450042725,0.4231261610984802
19 0.1455194503068924,-0.26840195059776306,0.4242856502532959

```

Para el procesamiento de estos datos se ha implementado un *script* en *Python* para leer y representar los datos mediante las librerías *matplotlib* y *mpl_toolkits*, el *script* puede verse en el Listado B.2. Como resultado de este *Script* obtenemos las gráficas mostradas en la Sección-6.4. Se puede observar en la Figura B.2 el resultado con una mayor resolución. La trayectoria azul es el ejercicio definido por el terapeuta, las trayectorias rojas son usando la aplicación sin *feedback* y las trayectorias verdes usando la aplicación de sistema. Los resultados han sido comentados en la Sección-6.4.

Listado B.2: Script para dibujar los resultados de la experimentación

```

1
2 import numpy as np
3 import csv
4 import os
5 from mpl_toolkits.mplot3d import Axes3D
6 import matplotlib.pyplot as plt
7
8 def readAndDraw(filename, color, linewidth, ax):
9     x_points = np.array([])
10    y_points = np.array([])
11    z_points = np.array([])
12
13    with open(filename, newline='') as File:
14        reader = csv.reader(File)
15        for row in reader:
16            x_points=np.append(x_points, float(row[0]))
17            y_points=np.append(y_points, float(row[1]))
18            z_points=np.append(z_points, float(row[2]))
19
20    ax.plot(x_points, z_points, y_points, color=color, linewidth=linewidth )
21
22 if __name__ == "__main__":
23
24    fig = plt.figure()
25    ax = Axes3D(fig)
26
27    # Ejercicio definido por el terapeuta
28    readAndDraw(filename='ejercicio.csv',
29                color='b',linewidth=3.0, ax=ax)
30
31    # Trayectorias sin feedback
32    for archivo in os.listdir('no-feedback'):
33        readAndDraw(filename='no-feedback/' + format(archivo),
34                    color='r',linewidth=1.0, ax=ax)
35
36    # Trayectorias con feedback
37    for archivo in os.listdir('feedback'):
38        readAndDraw(filename='feedback/' + format(archivo),
39                    color='g',linewidth=1.0, ax=ax)
40
41    ax.set_xlabel('X_axis')
42    ax.set_ylabel('Z_axis')
43    ax.set_zlabel('Y_axis')
44
45    plt.show()

```

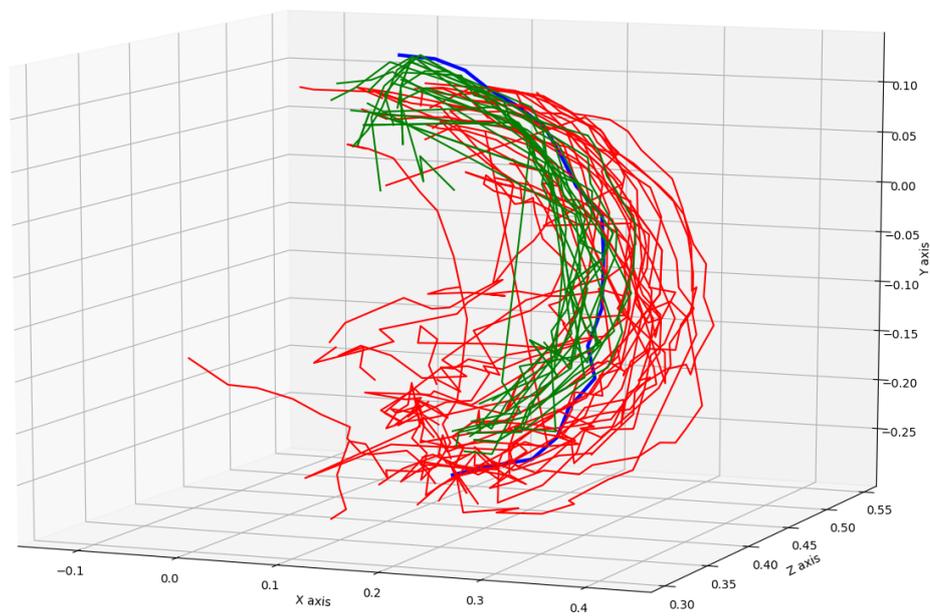


Figura B.2: Resultados de la experimentación con sujetos de prueba. Rojo sin *feedback*, verde con *feedback* y azul el ejercicio definido