



16.30 PROYECTO FINAL DE CARRERA
INSTITUTO TECNOLÓGICO DE BUENOS AIRES

***BrainWalkVR: aplicación en realidad
virtual para el entrenamiento
físico-funcional de los pacientes con la
Enfermedad de Parkinson***

Alumna:
Jungmin KOH

Tutor:
Ing. Santiago BESTANI

Co-tutor:
Dr. Fabian CRESMASCHI

Asesor:
Ing. Santiago RACCA

Fecha de entrega: 13 de noviembre de 2023

Resumen

La enfermedad de Parkinson (EP) es un trastorno neurodegenerativo crónico y progresivo que afecta los circuitos motores en el cerebro, resultando en síntomas como lentitud en los movimientos, rigidez, temblores y problemas de coordinación. Además, los pacientes pueden experimentar dificultades cognitivas y todo el conjunto de complicaciones culminan en una marcada alteración en la marcha. Los tratamientos farmacológico y quirúrgico que existen en la actualidad no mejoran significativamente el trastorno de la marcha, lo cual impacta en la calidad de vida de los pacientes.

En este proyecto final de grado, se diseñó una aplicación de entrenamiento en realidad virtual (RV) llamada *BrainWalkVR*, que sirve como herramienta de rehabilitación física y funcional en pacientes con Parkinson. Se llevó a cabo una fase de testeos con 11 participantes sanos, para evaluar la usabilidad del sistema, el grado de síntomas asociados con el *cybersickness*, y el rendimiento físico que logra exigir una sesión de entrenamiento de *BrainWalkVR*. Los resultados indicaron que la aplicación es altamente usable debido a la facilidad de aprender y al alto grado de satisfacción en los usuarios. Además, los participantes presentaron bajo nivel de síntomas de *cybersickness*, y el análisis de datos inerciales reveló que los participantes realizaron una cantidad significativa de pasos y giros, recorriendo una distancia considerable, durante el entrenamiento. El proyecto concluye que la aplicación *BrainWalkVR* tiene potencial para complementar los tratamientos de rehabilitación convencionales de la EP, y tendrá un impacto positivo en la calidad de vida de los pacientes al mejorar su movilidad y coordinación. A futuro, se sugiere realizar pruebas adicionales con un group más amplio de participantes y colaborar con profesionales de la salud para integrar la aplicación en la práctica clínica.

Agradecimientos

Quisiera expresar mi más profundo agradecimiento a todas las personas que me han acompañado durante la realización de mi proyecto final de grado.

Primero, les doy gracias al Dr. Fabián Cresmaschi y al equipo de Sensus 3D, Santiago Bestani y Santiago Racca, quienes me brindaron valiosa orientación y apoyo a lo largo de esta investigación. Su experiencia y dedicación fueron fundamentales para la realización de este proyecto.

También agradezco a mis profesores de la institución, Marcos Crespo, Sandra Wray, y Giuliana Antonella, quienes me brindaron su infinita ayuda durante la fase de testeo, proporcionando los materiales y el espacio necesario, y ayudándome a comprender a fondo las complejas señales inerciales. Su generosidad y colaboración fueron esenciales para seguir adelante con mi proyecto.

No quiero dejar de mencionar a los participantes que formaron parte de la fase de testeo de este proyecto, que sin su compromiso y contribuciones, no habría sido posible obtener los resultados relevantes que respaldan este proyecto.

Finalmente, quiero dar las gracias a mi madre, Lucía, y a mi novio, Franco, que me estuvieron acompañando en cada etapa de este proyecto. También a mi padre, quien me brindó apoyo constante a pesar de la distancia que nos separa. A todos ustedes, les agradezco por su apoyo, amor y comprensión que me impulsaron a superar los desafíos e hicieron posible la culminación de este proyecto.

Índice

1. Presentación	6
1.1. Introducción	6
1.2. Objetivos	7
2. Marco teórico	8
2.1. El Sistema Nervioso	8
2.1.1. Estructura anatómica	8
2.1.2. Vías de control de los Ganglios Basales	12
2.2. Enfermedad de Parkinson	14
2.2.1. Fisiopatología	14
2.2.2. Diagnóstico	16
2.2.3. Tratamientos en la actualidad	17
2.2.4. Rehabilitación motora y cognitiva	18
2.3. Realidad Virtual	21
2.3.1. Definición	21
2.3.2. <i>Cybersickness</i>	23
2.3.3. Aplicaciones en la Enfermedad de Parkinson	24
2.3.4. Situación actual en Argentina	27
2.4. Unidad de Medición Inercial (IMU)	28
2.4.1. Aceleración lineal y velocidad angular	29
2.4.2. Ángulos de Euler	30
3. Materiales y métodos	31
3.1. Hardware	31
3.1.1. Dispositivo de realidad virtual	31
3.1.2. Modo de uso	32
3.2. Software	33
3.2.1. <i>Unity</i>	33
3.2.2. Flujo del Programa	35
3.2.3. Escena de Inicio	39
3.2.4. Escena principal	41
3.2.5. Escena Final	45
3.3. Fase de testeo	46
3.3.1. Protocolo	46
3.3.2. Sistema de Escala de Usabilidad (SUS)	48
3.3.3. Cuestionario de Cibermareo en Realidad Virtual (CSQ-VR)	50
3.3.4. Sensor inercial Xsens DOT	51
3.4. Procesamiento de señales inerciales	52
3.4.1. Cálculo del recorrido total	53
3.4.2. Cálculo de la cantidad de pasos	54
3.4.3. Cálculo de la cantidad de giros	56

4. Resultados	59
4.1. Usabilidad del sistema	59
4.2. Grado de <i>cybersickness</i>	61
4.3. Mediciones inerciales	64
5. Discusión	71
5.1. Usabilidad del sistema desde la perspectiva de los usuarios	71
5.2. Grado de <i>cybersickness</i> experimentado por los usuarios	72
5.3. Rendimiento físico de los usuarios	74
6. Conclusiones	76
7. Bibliografía	78
8. Glosario	86
9. Anexo	88
9.1. <i>Scripts</i> del programa en Unity	88
9.2. Fase de testeo	112

1. Presentación

1.1. Introducción

La enfermedad de Parkinson (EP) es un trastorno neurodegenerativo crónico y progresivo, que se acompaña de una carga importante tanto física como psicológica y social en el sujeto que lo padece. Junto al aumento de la esperanza de vida mundial y el consecuente envejecimiento poblacional, la prevalencia global de la EP está incrementando: según la Organización Mundial de Salud (OMS), la cantidad de pacientes con Parkinson se duplicará con respecto a la cifra de 6,2 millones en 2015, a 12,9 millones en 2040 [1]. En Argentina, actualmente viven cien mil personas con dicha enfermedad y es el segundo trastorno neurodegenerativo más frecuente en el país [2].

A pesar de los últimos avances tanto en el diagnóstico como en el tratamiento, la enfermedad sigue siendo incurable y muchos de sus síntomas responden parcialmente a los tratamientos existentes en la actualidad. Por ejemplo, el método farmacológico con Levodopa, además de ser “máximamente invasivo” al afectar todo el organismo, va perdiendo su eficacia con el paso del tiempo, y puede producir efectos secundarios que son tanto o más severos que los síntomas originales de la EP [3].

Otro tratamiento moderno es la Estimulación Cerebral Profunda (ECP), aplicada por primera vez en 1987 para la EP, donde se estimulan determinadas áreas del cerebro a través de la colocación de los electrodos [4]. Si bien esta técnica es mínimamente invasiva y es eficaz en la reducción de temblores, no mejora el trastorno de la marcha que conlleva la rigidez y la bradicinesia, y también es poco eficaz para los síntomas no motores como el deterioro cognitivo y la dificultad en habla [5]. Dado este contexto, se vuelve fundamental desarrollar un método fisioterapéutico que permita mejorar los aspectos motores y cognitivos complejos, que no pueden tratar la medicación o la operación en solitario.

La realidad virtual (RV) es una tecnología que permite crear un entorno inmersivo virtual, donde la persona puede ver e interactuar con distintos objetos a su alrededor. Además de ofrecer un gran rango de experiencias diferentes al usuario, la RV es capaz de generar una sensación de presencia y de inmersión, de manera tal que la persona se pueda sentir parte de ese mundo virtual imaginario. Esto se logra gracias a los avances en hardware y software, que permiten desarrollar una experiencia de RV cada día más realista. Otra razón por la que esta tecnología se ha vuelto más popular es por la disminución de los precios de dispositivos de RV, que permite al público acceder más fácilmente a la RV [6]. Por ende, la realidad virtual está emergiendo como una de las herramientas potenciales para diversas aplicaciones.

En el presente trabajo, se desarrolla una aplicación de entrenamiento en RV, llamada *Brain-Walk VR*. La idea del proyecto se basa en un estudio de investigación en la Universidad de Southern California, *Design and Development of a Virtual Reality-Based Mobility Training Game for People With Parkinson's Disease* [7], donde se desarrolló un software que permite a los pacientes con Parkinson practicar técnicas como girar y caminar en superficie, mientras resuelven problemas cognitivos. Se decidió trabajar con la idea de este paper ya que esta aplicación de entrenamiento permite tratar en conjunto el trastorno de la marcha y el deterioro cognitivo que trae la EP, y será una herramienta útil para la rehabilitación física-funcional para los pacientes. Además, la creación

de un entorno virtual inmersivo que sea visualmente atractivo y permite interactuarse con el usuario, posiblemente aumentará la motivación de los pacientes para realizar el tratamiento en forma continua. Otra particularidad de la aplicación es que permite al paciente no solamente practicar a caminar, sino también los movimientos de rotación. De esta manera, el presente proyecto busca complementar los tratamientos de rehabilitación que existen en la actualidad, para mejorar la calidad de vida de los pacientes con Parkinson.

1.2. Objetivos

El objetivo principal del trabajo es desarrollar *BrainWalkVR*, una aplicación de entrenamiento en RV inmersiva, para fomentar la instauración de los ejercicios físicos en los pacientes con EP. El software se plantea con lineamientos de gamificación, en donde el rendimiento de la práctica se ve reflejado en los resultados de la sesión del entrenamiento, con el fin de objetivar el rendimiento y construir un entorno *user-friendly*. Para la realización del proyecto, se propusieron los siguientes objetivos de mínima y de máxima:

Objetivos de mínima:

- Desarrollar un simulador de realidad virtual como herramienta de rehabilitación de pacientes con enfermedad de Parkinson.
- Validación interna continua.
- Evaluar los efectos adversos de realidad virtual: *cybersickness*.

Objetivos de máxima:

- Desarrollar diferentes escenarios y ambientes para que el usuario pueda elegir según su preferencia, y que así se logre personalizar el tratamiento y aumentar sus motivaciones.
- Validación externa con la población sana, a través de los cuestionarios y las mediciones inerciales durante la fase de testeo.

2. Marco teórico

2.1. El Sistema Nervioso

2.1.1. Estructura anatómica

El sistema nervioso (SN) es el centro de comando del cuerpo humano, que controla los movimientos, pensamientos y emociones, y provoca reflejos ante un estímulo externo. Además, regula otros sistemas o procesos fisiológicos como la digestión y la respiración, y nos permite interpretar la información sensorial obtenida por los sentidos. La estructura del SN incluye el encéfalo, la médula espinal y los nervios periféricos. Los primeros dos forman el sistema nervioso central (SNC), mientras que el sistema nervioso periférico (SNP) se encarga de transportar mensajes de control del SNC al resto del cuerpo, a través de los nervios.

La unidad básica del SN es una neurona que tiene un cuerpo celular con su respectivo núcleo, y extensiones especiales llamadas axones y dendritas (Figura 2.1). Los conjuntos de axones se denominan nervios, y se extienden por todo el cuerpo construyendo el SNP. Cuando una neurona envía un mensaje a otra neurona, manda un impulso eléctrico que viaja a lo largo del axón, hasta que en la sinapsis, la señal se convierte en una señal química, liberando sustancias llamadas neurotransmisores. Los neurotransmisores transmiten la señal por la sinapsis hasta la dendrita de la neurona blanca, donde la señal química vuelve a convertirse en una eléctrica. De esta manera, los impulsos nerviosos generados en el SNC llegan a todas partes del cuerpo humano.

Además de la comunicación, en la sinapsis ocurre el fenómeno de la plasticidad neuronal o neuroplasticidad, donde el SN va modificando sus estructuras y conexiones neuronales, adaptándose rápidamente a las condiciones de su entorno. La neuroplasticidad es una característica fundamental del SN que nos permite adquirir nuevos conocimientos y capacidades tanto cognitivas como motoras [8].

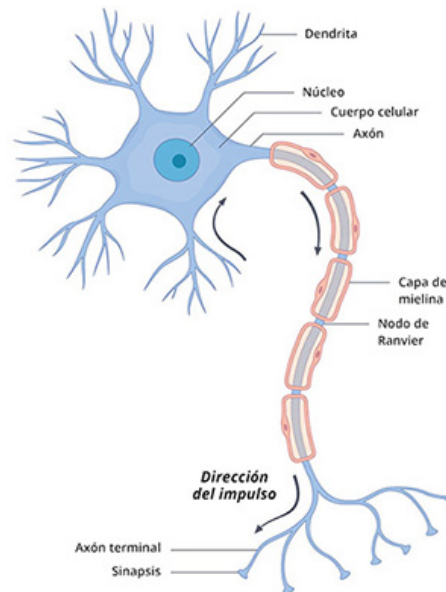


Figura 2.1: Partes de una neurona [9]

Retomando el enfoque en el SNC, el encéfalo está compuesto por tres partes principales: el cerebro (o telencéfalo), el cerebelo y el tronco encefálico (Figura 2.2). El cerebro está formado por los tejidos de sustancia gris y de sustancia blanca, que contienen los cuerpos celulares de las neuronas y las fibras nerviosas, respectivamente. La sustancia gris se encuentra en la superficie cortical, mientras que la sustancia blanca se encuentra en los tejidos más profundos, debido a que los axones se extienden para hacer sinapsis con las segundas neuronas del cuerpo (Figura 2.3). Si bien todos los componentes del SNC participan en el control motor, en el presente trabajo se pondrá el enfoque en el cerebro y en el tronco encefálico, que están mayormente involucrados en la fisiopatología de la enfermedad de Parkinson.

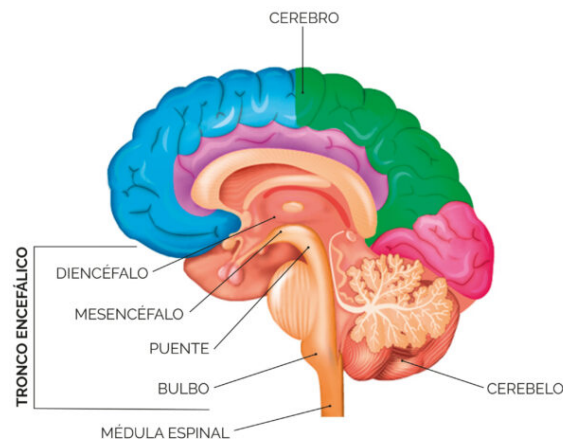


Figura 2.2: Partes del encéfalo [10]

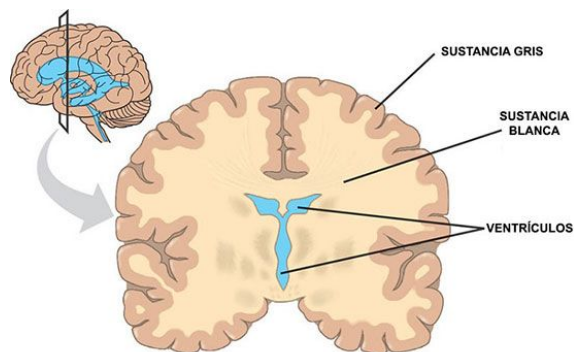


Figura 2.3: Sustancia gris y blanca del cerebro [11]

Los hemisferios cerebrales, derecho e izquierdo, se dividen en cuatro lóbulos principales: frontal, parietal, temporal y occipital (Figura 2.4). A su vez, la corteza cerebral se organiza en distintas áreas según sus funciones (Figura 2.5). Con respecto al surco central, las áreas responsables del control motor se agrupan en el lóbulo frontal, mientras que la corteza de los tres lóbulos restantes se encarga principalmente de la percepción e interpretación de la información sensitiva. Por esta razón, en muchas ocasiones el término «corteza frontal» se refiere al conjunto de áreas en el lóbulo frontal que regulan la planeación e iniciación de la actividad motora [12, 13].

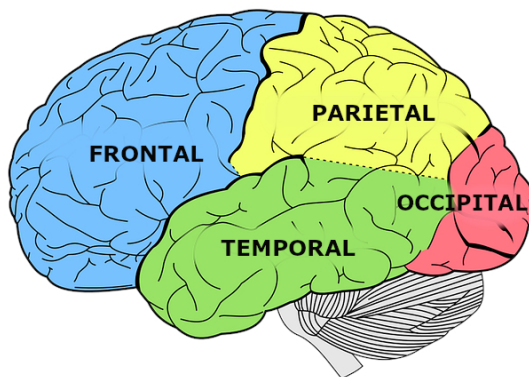


Figura 2.4: Lóbulos cerebrales [14]

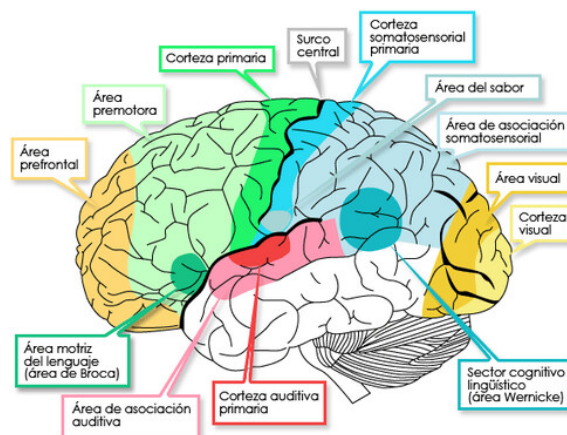


Figura 2.5: Áreas de la corteza cerebral

La corteza frontal está compuesta principalmente por las cortezas prefrontal, premotora y primaria. La corteza prefrontal es la porción que se involucra en varios procesos cognitivos complejos, como la toma de decisiones, el razonamiento, la expresión de la personalidad y la cognición social [15]. La corteza premotora, situada detrás de la prefrontal, es la zona esencial para la orientación sensorial de movimiento y para el control de los músculos proximales y del tronco [16]. Por último, la corteza motora primaria, que se encuentra en la parte anterior del surco central, se encarga de gestionar todas las acciones relacionadas con el control de los movimientos voluntarios [17].

Como se observó previamente en la Figura 2.2, el tronco encefálico está formado por el diencefalo, el mesencéfalo, el puente y el bulbo raquídeo. El diencefalo contiene una región llamada tálamo,

donde cada núcleo talámico recibe información de distintos orígenes: una vía sensitiva, el cerebelo o los ganglios basales. Según el origen de los impulsos nerviosos, el tálamo los proyectará al sector de la corteza cerebral que se relacione con dicha información. Por ejemplo, la vía óptica está asociada con el núcleo geniculado lateral del tálamo, por lo que su información será enviada a la corteza del lóbulo occipital, que corresponde al área visual [18]. En el diencefalo, también se ubican los núcleos subtalámicos cuya función es inhibir al tálamo en los circuitos motores. Por otro lado, el mesencéfalo tiene una porción de color negro llamada sustancia nigra, cuyas células producen dopamina, un neurotransmisor fundamental para la regulación del sistema sensitivomotor y el estado de ánimo. Tanto el tálamo como la sustancia nigra son partes del SNC que se involucran en la fisiopatología de la EP, junto a los ganglios basales [13].

Los ganglios basales o núcleos basales son un conjunto de masas de sustancia gris ubicadas dentro de cada hemisferio cerebral. Los núcleos basales son el cuerpo estriado, el núcleo amigdalino y el claustró, donde el cuerpo estriado incluye el núcleo caudado y el núcleo lenticular. A su vez, el núcleo lenticular se divide en el putamen y en el globo pálido con sus segmentos externo e interno [19] (Figura 2.6). Los núcleos subtalámicos y la sustancia nigra están estrechamente relacionados funcionalmente con los ganglios basales, y desempeñan un papel importante en el control de la postura y del movimiento voluntario. Por ejemplo, los receptores D1 y D2 del cuerpo estriado interactúan con la dopamina producida por la sustancia nigra, y activan distintas vías de control de los ganglios basales [13].

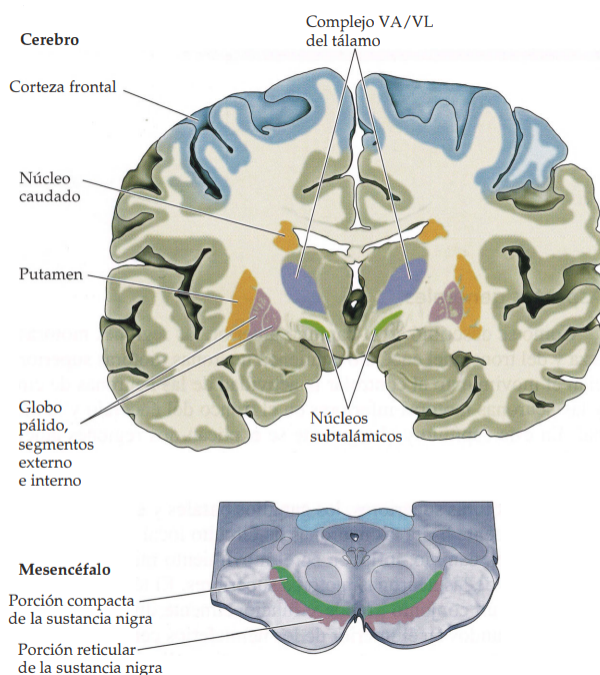


Figura 2.6: Localizaciones anatómicas de los ganglios basales y las estructuras relacionadas [20]

2.1.2. Vías de control de los Ganglios Basales

El control del movimiento del cuerpo humano se da como resultado de una alta interacción entre los siguientes cuatro subsistemas neurales: los circuitos locales de la médula espinal y del tronco encefálico, los sistemas moduladores descendentes, los ganglios basales y el cerebelo (Figura 2.7). Las motoneuronas superiores transportan información desde la corteza motora y el tronco del encéfalo hasta las motoneuronas inferiores en la médula espinal y el tronco encefálico. A su vez, las neuronas motoras inferiores inervan los músculos esqueléticos, cuyo movimiento estará dado por las órdenes del SNC.

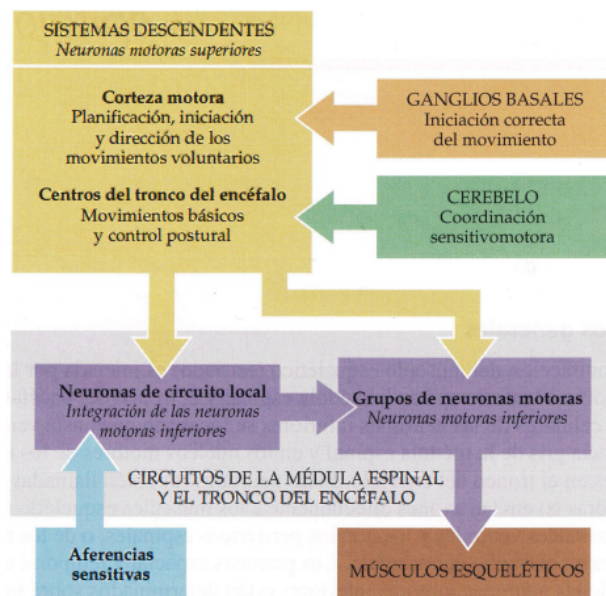


Figura 2.7: Estructuras neurales implicadas en el control del movimiento [20]

El cerebelo y los ganglios basales son circuitos con vías eferentes que no tienen acceso directo a las neuronas de circuito local ni a las motoneuronas inferiores; es decir, que no tienen un control explícito sobre la inervación de los músculos esqueléticos. Sin embargo, estos sistemas ejercen control sobre el movimiento mediante la regulación de la actividad de las motoneuronas superiores en la corteza cerebral y en el tronco encefálico. El cerebelo detecta la diferencia entre el movimiento que se intentó y el que realmente se realizó, y media la reducción de estos errores motores en tiempo real y a largo plazo. Mientras, los ganglios basales suprimen los movimientos no deseados y preparan los circuitos de las motoneuronas superiores para la iniciación de los movimientos voluntarios [20].

Como se mencionó previamente en la sección 2.1.1, la sustancia nigra tiene células productoras de dopamina, que a su vez interactúa con las neuronas del cuerpo estriado (núcleo caudado y putamen), que tienen los receptores a dopamina D1 y D2. Estos receptores modulan las vías directa e indirecta, respectivamente [13].

La vía directa es el circuito facilitador de movimiento ya que sirve para liberar las motoneuronas

superiores de la inhibición tónica. En cambio, la vía indirecta aumenta el nivel de inhibición tónica, por lo que dificulta la iniciación del movimiento. La dopamina de la sustancia nigra tiene un efecto excitatorio para los receptores D1 y para los receptores D2, un efecto inhibitor. Por esta razón, la presencia de dopamina activará la vía directa modulada por los D1, mientras que a falta de este neurotransmisor, se activará la vía indirecta regulada por los D2 (Figura 2.8).

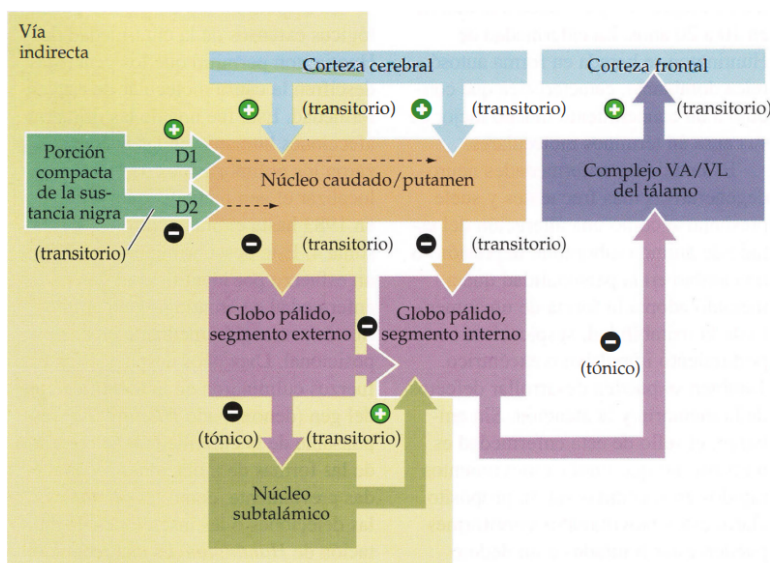


Figura 2.8: Vías indirecta y directa de los ganglios basales [20]

En la vía directa, las neuronas del cuerpo estriado son activadas por proyecciones de la corteza cerebral y por el efecto excitatorio de la dopamina. Luego, estas neuronas estriatales inhiben al globo pálido interno (GPI) que a su vez tiene inhibición tónica sobre los complejos ventral anterior y ventral lateral (VA/VL) del tálamo. Entonces, al inhibirse transitoriamente el GPI, el tálamo se desinhibe y envía aferencias excitadoras hacia la corteza frontal, facilitando la iniciación y el desarrollo del movimiento.

La vía indirecta se inicia cuando la corteza cerebral activa las neuronas inhibitorias del estriado o a falta de dopamina. Estas neuronas, a diferencia de las neuronas estriatales excitatorias, inhiben al globo pálido externo (GPE) que a su vez tiene inhibición tónica sobre el núcleo subtalámico. Entonces, al inhibirse transitoriamente el GPE, el subtálamo se desinhibe y envía aferencias excitadoras hacia el GPI. Como consecuencia, la inhibición tónica del GPI sobre el tálamo aumenta, conduciendo a que se bloquee el flujo de información motora hacia la corteza frontal.

En condiciones no patológicas, estas dos vías de señalización controlan las acciones del cuerpo contrarrestando entre sí. Sin embargo, cuando aparecen problemas asociados con los ganglios basales, este circuito complejo comienza a alterarse y da origen a los trastornos como la enfermedad de Parkinson.

2.2. Enfermedad de Parkinson

2.2.1. Fisiopatología

La enfermedad de Parkinson (EP) es un proceso neurodegenerativo crónico y progresivo que se caracteriza por sus síntomas tanto motores como no motores, que disminuyen la calidad de vida de las personas que lo padecen. Si bien se han identificado los posibles factores de riesgo como edad, herencia y exposición a toxinas, hasta el día de hoy la causa exacta de la enfermedad se desconoce [21]. La EP comienza cuando se degradan las neuronas productoras de dopamina en la sustancia nigra, lo cual desencadena una serie de alteraciones en las vías de los ganglios basales (Figura 2.9)

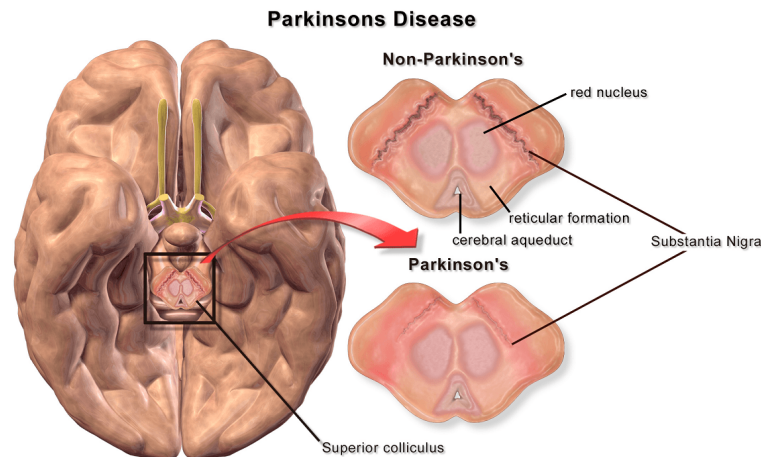


Figura 2.9: Deterioro de la sustancia nigra en un cerebro con la EP [22]

En el esquema de la Figura 2.10, se muestra cómo se altera el equilibrio entre las vías directa e indirecta de los ganglios basales en una persona con EP. A falta de dopamina, las neuronas estriatales de la vía directa se excitan menos, por lo que el GPI recibirá menos inhibición transitoria por parte del cuerpo estriado. Por otro lado, la disminución de la dopamina conduce a que las neuronas estriatales de la vía indirecta se activen más, aumentando su inhibición transitoria sobre el GPE. Entonces, el núcleo subtalámico recibirá menos inhibición por parte del GPE, lo cual aumentará sus aferencias excitatorias hacia el GPI. Como resultado, por la menor inhibición del cuerpo estriado en la vía directa y la mayor excitación del núcleo subtalámico en la vía indirecta, el GPI aumentará su inhibición tónica sobre los complejos VA/VL del tálamo. Todo este conjunto de alteraciones finalmente disminuye las aferencias excitatorias del tálamo hacia las motoneuronas superiores en la corteza frontal, generando como consecuencia una dificultad en la planeación e iniciación de la actividad motora del paciente. Por esta razón, la enfermedad de Parkinson se considera un trastorno hipocinético, donde la persona presenta expresiones faciales reducidas y movimientos restringidos.

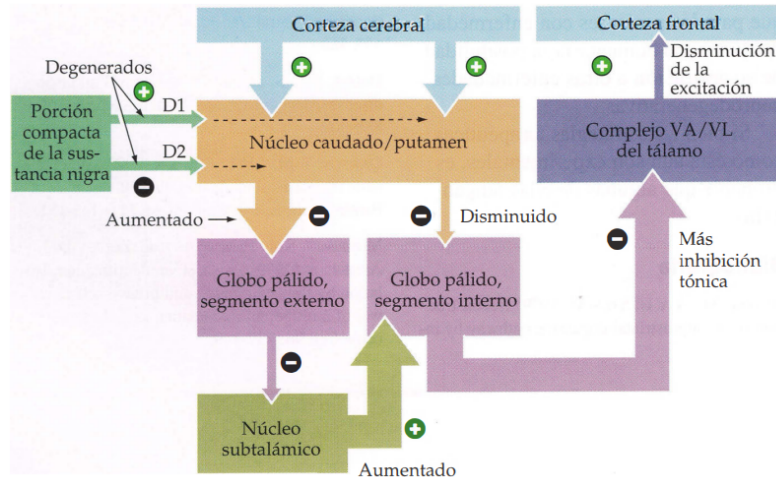


Figura 2.10: Alteración de los circuitos motores en los ganglios basales en la EP [20]

El disfuncionamiento de los circuitos de los ganglios basales produce una alteración en el control del movimiento, que se manifiesta por diversos síntomas motores en los pacientes. Uno de ellos es la bradicinesia, una reducción progresiva de velocidad de los movimientos a medida que continúan los movimientos. Otro es la acinesia, que es la dificultad para armar, iniciar y desarrollar un plan motor, mientras que los temblores son las sacudidas rítmicas que se producen en reposo, generalmente en las extremidades superiores. El paciente puede tener también una disartria hipocinética, donde presenta dificultad para decir las palabras debido a la falta de coordinación de los músculos del habla [23].

La alteración de las vías del control del movimiento en EP se manifiesta principalmente en los miembros inferiores. La rigidez muscular, que va aumentando con el progreso de la enfermedad, limita la amplitud del movimiento en todas las articulaciones del miembro inferior, por lo que un paciente con Parkinson muestra una forma de marcha muy característica y única de la enfermedad: la marcha parkinsoniana. Las características principales de una marcha parkinsoniana son la lentitud, los pasos cortos y arrastrados, el torso inclinado hacia adelante, y las extremidades superiores rígidas que no acompañan los movimientos de los pies y sufren de temblores. Este trastorno de la marcha, además de hacer perder la postura y el equilibrio al paciente, produce un fenómeno llamado la congelación de la marcha (CDM), que aparece como episodios intermitentes y cortos donde el paciente se encuentra incapaz de iniciar o mantener la locomoción, o realizar una rotación del cuerpo. El paciente también puede sufrir de festinación, donde el paciente comienza a caminar con pasos pequeños y cada vez acelera más, sin poder detenerse [24]. Estos síntomas de la marcha empeoran bajo situaciones de estrés o de presión, por ejemplo cuando la persona se encuentra en un lugar poblado [25]. Entonces, al no tener un control estable sobre la marcha, se aumenta el riesgo de caídas para el paciente, dificultando sus actividades diarias en vida cotidiana y por ende, disminuyendo su independencia y su calidad de vida.

Además de los síntomas motores, un paciente con EP sufre de un deterioro cognitivo que afecta a las funciones visoespaciales y ejecutivas, a la atención, a la memoria y al lenguaje. Por esta razón, la persona presenta dificultad para realizar tareas en secuencia que requieran atención o

planificación, lo cual se refleja en la vida cotidiana, cuando el paciente debe esforzarse para realizar actividades comunes como hacer café. Otro síntoma importante es la disminución de la habilidad de resolución de problemas: el paciente tiene menor flexibilidad cognitiva y menor capacidad para seleccionar estrategias para resolver problemas, planificarlas y ejecutarlas [26]. Por otro lado, el paciente puede presentar el llamado fenómeno de «tenerlo en la punta de la lengua», donde debe esforzarse para recordar palabras conocidas de vida cotidiana. Estos problemas con el lenguaje y con el habla se manifiestan más bajo situaciones de estrés o presión, igual que el trastorno de la marcha. Además, la función visoespacial del paciente se deteriora, por lo que tiene dificultad para percibir la distancia y la profundidad en el espacio, y a medida que la enfermedad avanza, puede tener problemas con el procesamiento de información sobre el entorno, acoplados con la demencia [27, 28].

Otra complicación que trae el daño cognitivo en la EP es el deterioro de *dual-tasking*, la capacidad de realizar dos tareas en forma simultánea. En una situación de *dual-tasking*, como por ejemplo caminar y hablar por teléfono al mismo tiempo, se requiere una interacción cognitivo-motora en el sistema nervioso, donde el cerebro procesa una combinación de dos tareas independientes y no relacionadas [7]. Debido al deterioro de su función ejecutiva, el paciente no podrá realizar esta interacción cognitivo-motora en forma efectiva, sea lo que sea la actividad que esté haciendo mientras camina, por lo que la velocidad y el desempeño de su marcha empeorarán [29, 30].

Por otro lado, la depresión es un trastorno mental que comúnmente se observa en pacientes con EP: la prevalencia de depresión en los pacientes es de un 50 %, aunque su relación fisiopatológica se desconoce [31]. A pesar de una tasa tan elevada, la importancia del tratamiento de la depresión en EP es muchas veces subestimada, ya que se considera como un problema menor en comparación con los síntomas motores. Sin embargo, es imprescindible realizar un asesoramiento al respecto para evitar una disminución de calidad de vida en pacientes, que luego puede acelerar el deterioro cognitivo y físico, discapacidad funcional rápida, y por lo tanto, el aumento de mortalidad.

2.2.2. Diagnóstico

Actualmente, la enfermedad de Parkinson es diagnosticada por un neurólogo, médico especializado en enfermedades del sistema nervioso. Como no hay una prueba particular para diagnosticar la enfermedad, se basa en los antecedentes médicos, una revisión de los síntomas y estudios por imágenes como la resonancia magnética, la ecografía cerebral o la tomografía por emisión de positrones. Sin embargo, lo que confirma definitivamente el diagnóstico de la enfermedad es ver si se observa una mejora significativa con administración de carbidopa y levodopa [21].

La enfermedad de Parkinson (EP) puede dividirse en las etapas inicial, intermedia y avanzada, y el médico puede diagnosticar en qué etapa se encuentra el paciente de acuerdo al grado de sus síntomas. Estas etapas se pueden definir por los estadios evolutivos según la escala de Hoehn y Yahr, que es una herramienta de clasificación para establecer la progresión de la EP, de acuerdo a la lateralización de la afectación (unilateral o bilateral) y alteraciones del equilibrio [32]:

- **Estadio 1:** Afectación exclusivamente unilateral, sin o con mínimo deterioro funcional.
- **Estadio 2:** Afectación bilateral o axial, sin alteración del equilibrio.

- **Estadio 3:** Afectación bilateral leve a moderada. Primer signo de alteración de los reflejos de enderezamiento. Siendo la respuesta a la fuerza de la gravedad, es evidente por la inestabilidad postural que se observa cuando el paciente da vuelta o pierde el equilibrio vertical con pies juntos y ojos cerrados. Limitación funcional en las actividades, pero físicamente independiente.
- **Estadio 4:** Discapacidad grave; aún capaz de caminar o de permanecer en pie sin ayuda.
- **Estadio 5:** Permanece en una silla de ruedas o encamado si no tiene ayuda.

Los estadios 1 a 2 corresponden a las etapas iniciales de la enfermedad; el estadio 3, a la etapa intermedia, y los 4 y 5, a las etapas avanzadas. Aunque se consideren pocas variables, la escala de Hoehn y Yahr es fácil de aplicar y se relaciona con el deterioro motor y la calidad de vida, por lo que se utiliza ampliamente en las investigaciones para seleccionar los sujetos de estudio.

En la etapa inicial, la persona camina con lentitud y pasos cortos, con disminución del braceo y cierta rigidez, que reducen su rango de movimiento y flexionan la postura del paciente. Estos síntomas suelen ser unilaterales al principio, por lo que se puede observar una asimetría significativa entre los movimientos de las extremidades.

En la etapa intermedia, muchos de los síntomas temporoespaciales de la etapa inicial progresan de manera bilateral, por lo que la asimetría disminuye pero la bradicinesia se vuelve más aguda y la automaticidad motora deteriora; por ejemplo, el paciente puede girar solamente en bloques debido a la fragmentación de su función motora. Por otro lado, en esta etapa aumenta el riesgo de caídas por el fenómeno de congelación de la marcha (CDM) y la festinación [24]. Por esta razón, es importante realizar los tratamientos de rehabilitación de la marcha en esta etapa intermedia, para que los pacientes puedan reducir el riesgo de caídas y mantener su independencia, y por ende tener mejor calidad de vida.

En la etapa avanzada, la persona ya tiene mayor riesgo de caídas por los episodios frecuentes de la CDM, y un control reducido de balance y de postura. Además, presentan fluctuaciones motoras y discinesia, que consiste en movimientos involuntarios, que agravan el trastorno de la marcha. En el momento en que los músculos pierdan su fuerza y su resistencia, disminuirá la capacidad motora del paciente y necesitará ayuda de una tecnología asistiva o una silla de ruedas para desplazarse o mantenerse en postura.

2.2.3. Tratamientos en la actualidad

El tratamiento farmacológico principal para la EP en la actualidad es levodopa, un fármaco que al ingresar al sistema nervioso se convierte en dopamina y recompensa la reducción de este neurotransmisor. Habitualmente, la administración de levodopa es acompañada por otro fármaco llamado carbidopa, que impide que la levodopa se descomponga antes de llegar al cerebro, y así permite una dosis inferior de levodopa. De esta manera, reducen los efectos secundarios de levodopa como náuseas, vómitos y presión arterial baja [33]. Si bien estos medicamentos disminuyen la rigidez muscular y los temblores al principio, su eficacia va decrementando con el tiempo por la tolerancia al fármaco, y puede generar consecuencias adversas como las discinesias: aproximadamente

el 50 % de los pacientes sufren de fluctuaciones involuntarias después de cinco años con el tratamiento de levodopa [34]. Además, el trastorno de la marcha no se suele mejorar significativamente con los tratamientos farmacológicos ni quirúrgicos [3]. Entonces, el tratamiento farmacológico con carbidopa y levodopa no resulta ser una terapia sostenible a largo plazo ni abarca todas las complejidades de la enfermedad, y posiblemente disminuya la calidad de vida de los pacientes a medida que pasen años de administración de medicamentos.

Otro tratamiento que existe en la actualidad es la Estimulación Cerebral Profunda (ECP), un tratamiento quirúrgico que consiste en localizar con alta precisión una determinada región del cerebro, como por ejemplo el núcleo subtalámico, y colocarle un electrodo que recibirá los impulsos eléctricos desde una batería externa. Esta estimulación corrige la actividad eléctrica anormal que causa trastornos del movimiento, y es una terapia altamente efectiva que permite reducir la dosis de levodopa y restaurar algunas partes de la función motora. Si bien este método mínimamente invasivo mejora los síntomas motores en las extremidades, el trastorno de la marcha o los problemas del equilibrio responden parcialmente al tratamiento. Además, el paciente debe someterse a unos protocolos de evaluación muy rigurosos y sistematizados antes de ser seleccionado; de hecho, la cirugía corresponde solamente al 20-30 % de los pacientes con Parkinson [3].

En definitiva, la enfermedad de Parkinson sigue siendo incurable y el paciente debe tratar sus síntomas a largo plazo a medida que la enfermedad progresa. Dado este contexto, es evidente que ni el tratamiento farmacológico ni los quirúrgicos en solitario tienen resultados suficientemente duraderos y efectivos, en particular para tratar el trastorno de la marcha y desacelerar el deterioro cognitivo que conlleva. Por este motivo, resulta fundamental desarrollar un método fisioterapéutico que permita tratar los síntomas de la enfermedad en forma sostenida y constante, con el fin de mejorar la calidad de vida de pacientes.

2.2.4. Rehabilitación motora y cognitiva

La rehabilitación es un conjunto de intervenciones diseñadas con el fin de optimizar el funcionamiento y reducir la discapacidad en individuos con condiciones de salud en interacción con su entorno [35]. En caso de los pacientes con Parkinson, la rehabilitación cobra mayor importancia ya que debe tratar una complejidad de síntomas tanto motores como no motores. Además, al ser una enfermedad crónica sin cura, es importante que el paciente reciba un tratamiento de rehabilitación consistente, para mantener o recuperar su independencia y por ende, mejorar su calidad de vida. El día de hoy, los tratamientos de rehabilitación para la EP que se realizan son principalmente la fisioterapia y las actividades de estimulación cognitiva.

La fisioterapia o terapia física es un tipo de tratamiento no farmacológico que incluye ejercicios físicos, masajes u otros métodos basados en estímulos físicos como calor, frío o estimulación eléctrica, que ayuden a optimizar la movilidad de las personas o aliviar su dolor [36]. La fisioterapia cobra mucha importancia en la rehabilitación para la EP, ya que su tratamiento se basa principalmente en las actividades físicas que fomentan la neuroplasticidad y la movilidad sostenible. En varias investigaciones previas, se demostró que los ejercicios físicos promueven la plasticidad neuronal al aumentar la fuerza de la sinapsis. De la misma manera, las actividades físicas fomentan la neurotransmisión en los circuitos funcionales, y también producen cambios estructurales en la sustancia gris del cerebro, que se reflejan en mejoras de balance del paciente con Parkinson [37, 38]. Además,

los estudios sugieren que la fisioterapia reduce el estrés oxidativo crónico que daña las células y sus funciones en la EP, y estimula la síntesis de neurotransmisores y factores tróficos que tienen un papel importante en la comunicación intercelular [39].

Para optimizar el rendimiento del paciente, una fisioterapia para la EP debe estar diseñada acorde a los siguientes principios [40]:

- **Aprendizaje por objetivos:** el tratamiento debe consistir en una serie de sesiones que conducen a una mejora de desempeño (entrenamiento de la marcha o ejercicios posturales).
- **Neuroplasticidad dependiente de experiencia:** el tratamiento debe tener determinada intensidad, repetición, y cierto grado de dificultad que le dé complejidad a los ejercicios.
- **Entrenamiento aeróbico:** el tratamiento debe ser una actividad sostenida que incrementa la capacidad cardiorrespiratoria, el consumo de oxígeno y el flujo sanguíneo cerebral.
- **Estimulación cognitiva:** durante el tratamiento se debe trabajar la propiocepción y la atención, y también aumentar la motivación.
- **Viabilidad:** el tratamiento debe tener objetivos realistas, y el paciente debe tener un grado de medicación óptimo para realizar la terapia.

En la actualidad, existe una gran variedad de fisioterapias para la EP, entre las cuales están el estiramiento muscular, el entrenamiento de resistencia, los ejercicios de balance y postura, y el entrenamiento en cinta de correr. Cada una de estas actividades permite trabajar ciertos aspectos físicos; por ejemplo, el estiramiento ayuda a reducir la rigidez muscular que causa una postura encorvada muy característica de la enfermedad, mientras que los ejercicios de resistencia incrementan la fuerza muscular que va perdiendo el paciente a medida que progresa la enfermedad.

Para tratar el trastorno de la marcha en Parkinson, el entrenamiento en cinta de correr, *treadmill training* (TT), es una de las fisioterapias más efectivas y utilizadas en la actualidad. Este tratamiento es fundamental para reducir el riesgo de caídas y mejorar la caminata, porque permite aumentar la velocidad de la marcha, la cadencia (el número de pasos por unidad de tiempo) y la longitud de pasos que ha disminuido por la enfermedad [41]. Además, al ser un ejercicio aeróbico, fortalece la capacidad cardiorrespiratoria del paciente y el flujo sanguíneo del cerebro.

Sin embargo, el TT tiene ciertas limitaciones para abarcar las complicaciones de la enfermedad. La actividad consiste en caminar sobre la cinta de correr sin cambiar la dirección, por lo que es ajeno a los movimientos en la vida cotidiana, donde la persona debe caminar, girar y cambiar de direcciones constantemente. Además, el TT no puede ejercitar *dual-tasking* (DT) ya que el paciente no está realizando ninguna otra actividad más que caminar. Para complementar esta limitación, se comenzaron a incorporar nuevas tecnologías como la realidad virtual y realidad aumentada en el entrenamiento de la marcha, para que los pacientes puedan entrenar su DT. Por ejemplo, caminando sobre la cinta de correr, el paciente tiene que superar los obstáculos virtuales en la pantalla, para que de esta manera pueda mejorar tanto la marcha como su función ejecutiva, lo cual implicaría reducir el riesgo de caídas [42]. No obstante, el *treadmill training* sigue sin permitir ejercitar el movimiento de rotación, lo cual es fundamental para que el paciente pueda realizar actividades de vida cotidiana y tener mejor calidad de vida.

Aparte de las sesiones de fisioterapias propiamente dichas, los médicos dan énfasis sobre la importancia de realizar una rutina de ejercicios diaria para tratar los síntomas de EP. Por ejemplo, el Dr. Marcelo Merello, médico neurólogo y Jefe del servicio de Movimientos Anormales de la Fundación para la Lucha contra las Enfermedades Neurológicas de la Infancia (Fleni) escribió un libro llamado *Parkinson. Guía esencial para pacientes, familiares y cuidadores* [43], donde describe en detalle los ejercicios que pueden hacer los pacientes para la movilidad del cuello y del torso. El autor les da mucha importancia a ejercicios de rotación, inclinación y flexión, ya que mejoran la rigidez muscular que va empeorando a medida que progresa la enfermedad. Además, en una charla de 2017 el médico neurólogo Malco Rossi, especialista en Movimientos Anormales en Fleni, marcó que realizando ejercicios entre 15 y 30 minutos diariamente, pueden mejorar la fuerza muscular, el equilibrio, la marcha y la postura. Además, permite regular el peso y renovar el estado de ánimo, contribuye al buen descanso y las funciones cognitivas, y reduce o previene la fatiga, el desgano y los dolores. Incluso, disminuye el riesgo de enfermedades cardiovasculares y cerebrovasculares, de osteoporosis o fracturas relacionadas con las caídas [44]. Por esta razón, además de que el paciente esté en un programa de rehabilitación física con su especialista, es importante que se comprometa a realizar ejercicios diariamente para mejorar efectivamente sus síntomas.

Las capacidades cognitivas son aquellas habilidades por las que una persona puede llevar a cabo cualquier acción o tarea en el día a día: aprender, memorizar, hablar, prestar atención, razonar, entender, leer, entre otras. Como se mencionó en la sección 2.2.1, enfermedad de Parkinson deteriora la memoria, el lenguaje y la función ejecutiva del individuo, por lo que además de perjudicar la vida cotidiana del paciente, disminuye el rendimiento de las actividades e incrementa el riesgo de caídas durante la marcha, conduciendo a la pérdida de independencia. Por eso, las actividades de estimulación cognitiva resultan fundamentales para desacelerar el deterioro cognitivo del paciente y por ende, mejorar su calidad de vida [45].

Si bien no existe un ejercicio de estimulación cognitiva en particular para la EP, se sugieren varias actividades para trabajar diferentes capacidades cognitivas [46, 47, 48, 49].

- **Sopa de letras:** buscar las palabras ocultas en un grupo de letras. Se trabajan la atención (sostenida y selectiva) y/o el razonamiento (si las palabras están relacionadas).
- **Relacionar conceptos o imágenes:** emparejar aquellas palabras o imágenes que tengan relación entre sí. Se estimula el uso del lenguaje, razonamiento y memoria semántica.
- **Parejas por categorías:** formar parejas de palabras en función de una categoría. Se trabajan la atención, el razonamiento y la memoria.
- **Palabras repetidas:** señalar las palabras que aparecen repetidas. Se estimulan la atención selectiva y la memoria de trabajo.
- **Ordenar pasos de una actividad:** Se trabajan la planificación y el razonamiento.
- **Ejercicios espaciales:** movimientos en 2D de una figura en 3D, lectura de mapas, entre otros. Ayudan con la orientación, la visualización espacial, la memoria y la concentración.
- **Juegos:** juegos de cartas, juegos de estrategia (por ejemplo, ajedrez), juegos de trivia, juegos de reflejos. Se estimulan la memoria, la capacidad de resolución de problemas, la atención, el razonamiento y la coordinación.

Se puede observar que los ejercicios que entrenan la función ejecutiva se basan en ejercicios de iniciativa (por ejemplo, recordar palabras), de categorización, de seriación o de planificación de una actividad. Además, estos ejercicios de estimulación cognitiva ayudan a mejorar la atención y la memoria que se deterioran por la enfermedad de Parkinson. También es importante destacar que los juegos de estimulación cognitiva permiten a su vez ejercitar el cerebro en forma entretenida, llevando a que los pacientes aumenten su motivación por la rehabilitación cognitiva y adquieran más fácilmente los hábitos de ejercitar diariamente.

2.3. Realidad Virtual

La realidad virtual (RV) es una tecnología moderna que ha avanzado significativamente en los últimos años, permitiendo la creación de experiencias más realistas en un entorno imaginario virtual. Además de tener mejor calidad gracias a los avances en términos de hardware y software, esta tecnología se ha vuelto más accesible y asequible por la disminución de los precios de dispositivos de RV, llevando a que su popularidad crezca con el tiempo [6]. La tecnología de RV permite a los usuarios experimentar situaciones y entornos que de otra manera serían imposibles o peligrosos en la vida real, e interactuar con el mundo virtual de forma similar al entorno real. Gracias a esta característica única, la RV se está convirtiendo en una de las herramientas utilizadas en una amplia gama de campos, incluyendo educación, entretenimiento, simulación de entrenamientos militares y medicina [50].

2.3.1. Definición

La realidad virtual se define como un conjunto de técnicas informáticas que permiten crear un entorno de escenas y objetos simulados, en donde una persona puede desenvolverse y experimentar la sensación de presencia, y también interactuar con su alrededor. Para disfrutar de una experiencia de RV, el jugador debe colocarse el casco y sostener los controladores de mano como se muestra en la Figura 2.11. Los visores del casco bloquean el mundo exterior al jugador y le muestran un mundo tridimensional que crea una simulación desde el punto de vista en primera persona, donde el usuario puede recibir también feedback audiovisual al interactuar con su alrededor. Además, si existe un avatar que represente al usuario en el mundo virtual, la persona podrá ver sus movimientos a través de la animación de su avatar. Los controladores con sus respectivos botones y joystick le permiten al usuario realizar diferentes acciones en el programa, tales como agarrar y sostener un objeto.



Figura 2.11: Un hombre usando los dispositivos de RV. En la pantalla de la computadora se muestra el mundo tridimensional que ve el usuario a través del casco [51]

Una experiencia de RV se caracteriza por el sentido de inmersión y el sentido de presencia que produce en los espectadores. La inmersión se define como la capacidad de involucrar los cinco sentidos del usuario al entorno virtual, de manera tal que lo haga sentir como si estuviera físicamente presente en ese mundo virtual. Esto depende mayormente de la resolución visual, del diseño de paisaje, personajes y objetos que deben ser similares a la experiencia real, y también de los feedback sensoriales que recibe el usuario al interactuar con su alrededor [52]. Por ejemplo, si uno se encuentra en un bosque de RV, los movimientos de las hojas de árboles, el sonido de pájaros y las vibraciones simuladas del viento estimularán los sentidos visuales, auditivos y táctiles del usuario, de manera tal que lo hagan sentir como si estuviese en un bosque real.

Mientras que la inmersión se concentra en involucrar al sujeto sensorialmente en el ambiente virtual, la presencia o la sensación de estar tiene como objetivo producir en el usuario una sensación psicológica y emocional, de forma tal que se comprometa con el mundo virtual y se sienta parte de él. Esta ilusión no es cognitiva sino perceptiva, ya que la persona está consciente de que es una simulación, aunque los sistemas sensoriales responden a los estímulos del entorno virtual como si fueran reales. Siguiendo el ejemplo del párrafo anterior, el usuario en algún momento empezará a sentirse emocionalmente conectado con el bosque virtual, relajándose y disfrutando de la paz que trae la naturaleza.

En pocas palabras, el sentido de inmersión corresponde al grado con el cual los sentidos son convencidos para que uno sienta que está en un entorno físico distinto a la realidad, mientras que el sentido de presencia se refiere a cómo uno se siente emocional y psicológicamente comprometido con ese mundo virtual, independientemente de estar consciente de que es una ilusión [53]. Los sentidos de inmersión y de presencia están fuertemente relacionados, y al aumentar el nivel de inmersión, la persona sentirá más presencia en el entorno virtual en que se encuentra [54].

Existen tres niveles de inmersión en la RV: no inmersiva, semi-inmersiva e inmersiva. Una realidad virtual no inmersiva es la experiencia que se visualiza en una pantalla de computadora, donde el usuario interactúa indirectamente con el entorno virtual con el mouse, el teclado o con

los joysticks, como por ejemplo en los juegos de computadora. Acá la persona no se siente parte de la RV, ya que su campo de visión está limitado por la pantalla y tampoco interactúa directamente con el ambiente. En una realidad virtual semi-inmersiva, la persona puede explorar visualmente el mundo virtual a través de los visores de RV o la pantalla, pero sigue sin tener sensación de estar físicamente en ese entorno. Por último, en una realidad virtual inmersiva, el usuario puede interactuar directamente con su alrededor usando los controladores de RV y el casco, y recibir feedback audiovisual que genere la sensación de estar. Además, si el campo de visión es de 360 grados, la persona se sentirá más inmersiva y presente en el entorno, al ser similar al campo de visión en la vida real [55]. Por esta razón, el nivel de inmersión es fundamental para que el usuario tenga mayor sensación de estar, y pueda tener una experiencia realista y conectarse más al mundo virtual.

2.3.2. *Cybersickness*

El *cybersickness* o el cibermareo consiste en un conjunto de efectos indeseados que puede presentar una persona durante o después de su exposición a un ambiente virtual. Estos malestares pueden ser cefalea, náuseas, fatiga ocular, oscilaciones corporales, inestabilidad o ilusión de movimiento, que son referidos en general como una sensación de mareo [56]. La mayoría de los síntomas de *cybersickness* tiene duración corta y bajo riesgo, cuando la exposición del usuario a la RV es durante cortos períodos de tiempo. Sin embargo, su intensidad y duración pueden incrementarse cuando la exposición se prolongue [57]. El conjunto de síntomas de *cybersickness* se denominan también como Síntomas y Efectos Inducidos por la Realidad Virtual (VRISE), y es fundamental evaluarlos al testear una aplicación de RV ya que se observan en entre un 60 y 80 % de usuarios de RV y es un factor importante que disminuye el sentido de presencia [58].

El *cybersickness* ocurre como consecuencia de la desincronización de la información obtenida entre el sistema visual y el sistema vestibular del ser humano, que se encarga de mantener el equilibrio y la postura. Cuando uno se encuentra parado y observa imágenes con mucho movimiento o aceleraciones visuales a través del casco de RV, esto puede producir una disonancia que confunde su sentido de la orientación, ya que los movimientos de la escena no están sincronizados con su movimiento corporal [56]. Por ejemplo, cuando el jugador gira la cabeza hacia un lado para ver los objetos, no los verá inmediatamente como ocurre en la vida real, sino que tardaría unos milisegundos hasta que en los visores del casco aparezcan los objetos. Ahí es el momento de desincronización que puede causar mareos u otros síntomas de *cybersickness* al usuario.

Por esta razón, se puede decir que el *cybersickness* depende fuertemente de la latencia, que es el tiempo de demora entre lo que el sistema de RV entrega y lo que el cerebro espera percibir. Este tiempo de retraso se define por la frecuencia de actualización (*refresh rate*) del dispositivo, que es el número de veces por segundo que la imagen se actualiza en la pantalla. Por ejemplo, una pantalla de 60 Hz actualizará la pantalla 60 veces por segundo [59]. Entonces, al aumentar la frecuencia de actualización, también mejorará la suavidad y la velocidad de cambio del movimiento en la pantalla, por lo que habrá menor latencia y por ende, el usuario sufrirá de *cybersickness* con menor intensidad.

Los Síntomas y Efectos Inducidos por la Realidad Virtual (VRISE) se pueden evaluar por los cuestionarios que completan los usuarios luego de su exposición al entorno virtual. Uno de ellos es

el Cuestionario de malestar asociado al simulador (SSQ), basado en la escala de Likert de cuatro puntos, que evalúa los efectos producidos por un simulador de vuelo. Sin embargo, los malestares por simulador no son equivalentes al *cybersickness*, ya que las náuseas y la desorientación son más frecuentes e intensas en *cybersickness* [60]. Más, se demostró que el SSQ no cubre las propiedades psicométricas adecuadas para evaluar el *cybersickness* en RV. Si bien existe una variante del SSQ llamada Cuestionario de malestar asociado a realidad virtual (VRSQ), este cuestionario tiene ciertas limitaciones tales como dificultad en interpretar sus resultados y el hecho de no abarcar preguntas relacionadas a náuseas, que es uno de los síntomas principales del *cybersickness* [61]. Por esta razón, en el presente trabajo se decidió evaluar los VRSE con el Cuestionario de Cíbermareo en Realidad Virtual (CSQ-VR), un cuestionario basado en la escala de Likert de siete puntos que abarca los síntomas principales del *cybersickness*: las náuseas, la desorientación y los síntomas oculomotores [62]. El CSQ-VR, además de tener propiedades psicométricas superiores al SSQ y VRSQ, demostró ser altamente sensible y específico en detectar la intensidad de los VRSE.

2.3.3. Aplicaciones en la Enfermedad de Parkinson

Como se explicó anteriormente, la enfermedad de Parkinson genera deterioro tanto cognitivo como motor en los pacientes, por lo que es importante realizar terapia física y actividades de estimulación cognitiva para desacelerar el progreso de la enfermedad y mejorar la calidad de vida de los pacientes. Gracias a su capacidad de diseñar un entorno interactivo y visualmente atractivo, la realidad virtual ha emergido como una herramienta potencial en el área de rehabilitación, especialmente para la EP, donde se requiere diseñar tratamientos dinámicos y entretenidos para ejercitar la movilidad y la función ejecutiva de los pacientes.

La idea de usar la RV para tratar el trastorno de la marcha en Parkinson surgió por primera vez en 1998, cuando el doctor Thomas J. Riess presentó un dispositivo terapéutico basado en realidad aumentada [63]. Después de esta propuesta, no hubo otros estudios de investigación que utilizaron la RV para el entrenamiento de la marcha parkinsoniana hasta el año 2010, cuando por primera vez se diseñó un *treadmill training* (TT) combinado con la RV (TT+RV). En este estudio, los pacientes tenían que caminar sobre la cinta de correr, mientras que superaban obstáculos físicos en un entorno virtual no inmersivo que se mostraba en una pantalla (Figura 2.12). Como resultado, después de recibir 18 sesiones del entrenamiento, los pacientes aumentaron su rapidez de marcha significativamente. Además, su capacidad de *dual-tasking* mejoró ya que disminuyó la variabilidad entre cuando marchan normalmente y cuando marchan bajo situaciones de *dual-tasking*. De esta manera, se demostró que el tratamiento TT+RV es capaz de mejorar tanto el trastorno de la marcha como el deterioro de funciones cognitivas en pacientes con Parkinson [42].



Figura 2.12: Primer tratamiento en cinta de caminar acoplado a realidad virtual para pacientes con Parkinson [42]

Más adelante, entre 2012 y 2019, se realizaron estudios de investigación que cuantificaron la efectividad de RV en el tratamiento del trastorno de la marcha en Parkinson. El entrenamiento con RV tuvo un efecto significativo sobre la mejora de balance [64, 65, 66, 67, 68, 69, 70, 71, 72], de movilidad [64, 67, 68, 70, 72, 73, 74], y de longitud de paso y de zancada [65, 73, 74, 75]. Además de tratar los síntomas motores, la aplicación de RV en rehabilitación aumentó la calidad de vida de los pacientes [64, 75, 73, 74, 76, 77], les dio mayor nivel de confianza [65, 75, 74] y mejoró sus síntomas neuropsiquiátricos [70, 73, 76, 78].

Otra utilidad de RV es la posibilidad de simular situaciones que podrían provocar la congelación de la marcha (CDM) en vida real, para estudiar cómo se manifiesta el trastorno de la marcha del paciente en esas situaciones. Efectivamente, el trastorno de la marcha suele empeorar cuando el paciente se encuentra en una situación bajo presión o en un ambiente poblado. En un estudio de investigación en 2023, se estudiaron los parámetros de la marcha tales como el ritmo de paso, la asimetría y el control postural y la festinación, mientras que los pacientes caminaban en un espacio físico abierto, estando en un entorno virtual inmersivo que simulaba diferentes situaciones: estar en un espacio vacío, pasar por una puerta, caminar por un pasillo y estar en un espacio público con gente (Figura 2.13). Estas simulaciones de RV fueron efectivas en recrear los momentos provocadores de congelación de la marcha (CDM), ya que se observó que el trastorno de la marcha de los pacientes empeoró cuando caminaron con RV, en comparación con la prueba sin RV [25]. Entonces, al simular situaciones que son difíciles de replicar en una clínica de rehabilitación, la RV permite a los terapeutas ofrecer un tratamiento más aplicado a la vida real, donde se puede estudiar mejor su fenómeno de CDM y evaluar cómo se puede reducir el riesgo de caídas en su vida diaria.

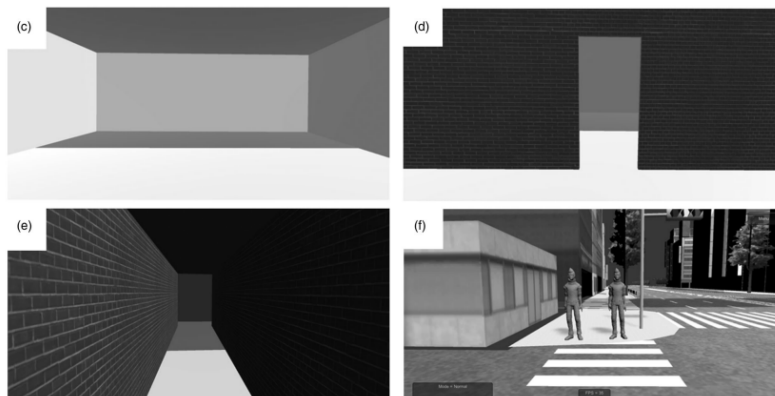


Figura 2.13: Distintos ambientes simulados en realidad virtual para las pruebas de marcha [25]

Por otro lado, la realidad virtual se aplica también para mejorar las funciones cognitivas que han deteriorado por la enfermedad. En un estudio de investigación [79], se estudiaron los impactos que un sistema terapéutico de RV semi-inmersiva llamado *BTS Nirvana* tiene sobre las funciones cognitivas de los pacientes. Después de realizar tres sesiones de entrenamiento por semana durante ocho semanas, se observó que los pacientes que realizaron el tratamiento con *BTS Nirvana* mejoraron sus funciones cognitivas, más que aquellos pacientes que estuvieron con los tratamientos convencionales durante el mismo período de tiempo. Además, se realizó una revisión sobre los estudios que trabajaron con la RV para tratar el deterioro cognitivo, y se descubrió que de los diez estudios seleccionados, la mitad concluyeron a que los *exergames*, combinación de juegos y ejercicios físicos, tenían resultados positivos sobre las funciones cognitivas, mientras que la otra mitad mostraron una mejora significativa en funciones ejecutivas en particular. Por esta razón, se puede decir que la RV, además de ejercitar la movilidad, permite desacelerar el deterioro cognitivo que trae la enfermedad de Parkinson.

La idea del presente trabajo se basa en otro estudio de investigación, donde se diseña una aplicación en RV inmersiva para el entrenamiento de pacientes con Parkinson [7]. El sistema se basa en un juego de ahorcado, donde el paciente debe completar una serie de palabras incompletas juntando piezas de letras distribuidas en un espacio abierto. De esta manera, el paciente podrá caminar y practicar giros, y también resolver problemas, trabajando en conjunto la movilidad y la función cognitiva. En el paper, se demostró que el sistema era altamente usable, tanto para el paciente como el terapeuta que supervisa el entrenamiento, y también aumentaba significativamente la motivación de los pacientes, lo cual es fundamental para que sigan realizando los tratamientos a largo plazo. Por lo tanto, se puede decir que esta aplicación en realidad virtual es una herramienta potencial en área de rehabilitación para ofrecerles a los pacientes un tratamiento efectivo y más entretenido.

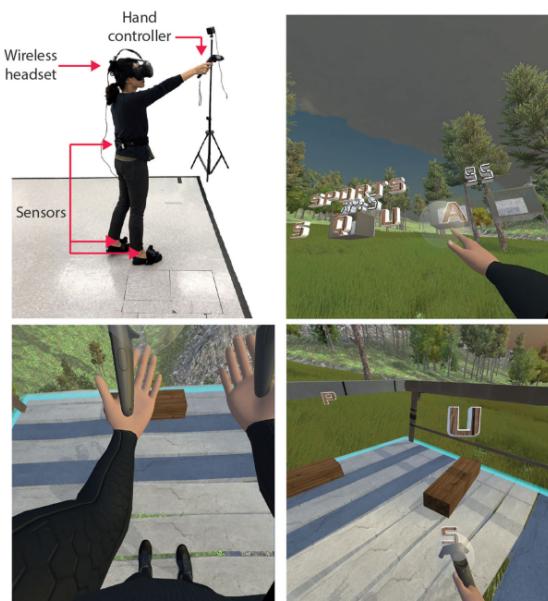


Figura 2.14: Pruebas y escenas de la aplicación de entrenamiento en el paper [7]

2.3.4. Situación actual en Argentina

Actualmente en Argentina, la RV es una herramienta relativamente nueva que recién está surgiendo en área de rehabilitación. Sin embargo, se comenzaron a desarrollar varios proyectos que buscan aplicar dicha tecnología para la rehabilitación de los pacientes. Una empresa cordobesa llamada Motmi ofrece una plataforma de rehabilitación virtual no inmersiva, que trackea el cuerpo del paciente con tecnología de detección y reconocimiento de movimiento, y lo representa con un avatar dentro de un videojuego que tiene un objetivo terapéutico particular [80]. En la Universidad Nacional Villa María, dos estudiantes de la Licenciatura en Terapia Ocupacional crearon una aplicación de RV para la rehabilitación de movimiento articular de muñeca. Usando los sensores de detección de movimiento, se trackean los movimientos de manos durante los juegos de RV, y de esta manera, el paciente puede tener programa de rehabilitación de calidad en forma más entretenida [81].

Por otro lado, la realidad virtual ya está en uso activo en los centros de rehabilitación en Argentina. Actualmente, en el Laboratorio de Rehabilitación Virtual de la Universidad de San Martín, se utiliza la plataforma de la empresa Motmi para rehabilitar a los pacientes con dificultades neuromotoras. En 2019, el Centro de Evaluación y Tratamiento con Tecnologías Aplicadas a la Rehabilitación (CETTAR) en Fleni Escobar incorporó nuevos equipamientos para la evaluación y el tratamiento de niños y adultos en materia de trastornos del balance y equilibrio, disfunciones motoras y compromiso de las funciones cognitivas. Entre esta nueva tecnología adquirida, se destaca el equipo C-Mill VR+ de la empresa *Hocoma Motek*, que consiste en una cinta de correr acoplada a un sistema de realidad extendida (realidad aumentada y virtual), para rehabilitación de marcha y de balance [82].

En la Figura 2.15, se puede ver el equipo C-Mill VR+ que tiene integradas una pantalla donde se

visualiza el entorno virtual, y la cinta de correr de 1×3 metros, sobre la cual aparecen proyecciones de realidad aumentada luminosas que guíen al paciente para dar pasos de cierta longitud. Por otro lado, tiene un soporte parcial de peso para la seguridad, y ofrece feedback visual con cámaras y mide parámetros de marcha en tiempo real con plataformas de fuerza en la cinta. Estos datos pueden ser utilizados por los especialistas para luego analizar el rendimiento del tratamiento y evaluar el mejoramiento en marcha y balance del paciente [83].



Figura 2.15: Componentes del equipo C-Mill VR+ de la empresa *Hocoma Motek* [83]

Es importante destacar que la actualización tecnológica de Fleni tuvo un aporte total de \$30.412.087 lo cual fue un monto significativo. Por otro lado, el equipo C-Mill VR+ tiene precio de \$215.000 en una plataforma de venta online de Australia, lo cual equivale a U\$D 136.900 aproximadamente [84]. Si bien consiste en una herramienta efectiva para rehabilitación de marcha, no todas las instituciones médicas tienen acceso económico a estos equipamientos complejos de alto costo, particularmente en países de Latinoamérica donde el ingreso promedio de la población es más bajo. Además, el C-Mill VR+ no ofrece una experiencia totalmente inmersiva por lo que el paciente tendrá menor sensación de presencia. Por lo tanto, se vuelve importante desarrollar un método de rehabilitación para la marcha con realidad virtual, que sea económicamente accesible y que al mismo tiempo permita ofrecer al paciente un tratamiento más dinámico y entretenido, de manera tal que se sienta más motivado para continuar con el tratamiento a largo plazo.

2.4. Unidad de Medición Inercial (IMU)

En la rehabilitación de la enfermedad de Parkinson, los sensores inerciales se utilizan muchas veces para medir parámetros del ciclo de marcha tales como longitud de paso, cadencia, y velocidad de la marcha, para entender la marcha parkinsoniana y luego determinar la eficacia de los tratamientos. Además de ser un método no invasivo, estos sensores son *wearable* por lo que no limitan la variedad y el rango de movimiento de la persona al realizar la prueba [85].

Una Unidad de Medición Inercial (IMU) es un sensor compuesto por acelerómetros y giróscopos, que procesa sus mediciones con microcontroladores y transmite sus datos a través de Bluetooth. Los acelerómetros sensan la aceleración lineal, mientras que los giróscopos, la velocidad angular del movimiento. Además, devuelve el cambio de orientación y otras fuerzas gravitatorias, que son útiles a la hora de analizar el movimiento de una persona.

2.4.1. Aceleración lineal y velocidad angular

La aceleración lineal y la velocidad angular son propiedades físicas vectoriales, que se pueden representar en un sistema de coordenadas ortogonal de tres ejes x, y, z (Figura 2.16). La magnitud de un vector se define por la distancia entre su punto inicial y su punto final, y su orientación y sentido se representan por una flecha.

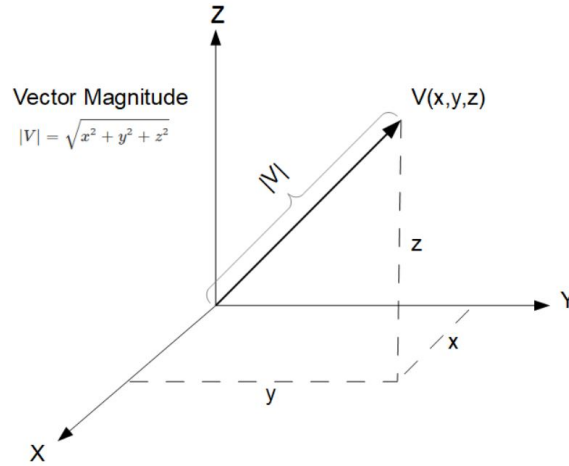


Figura 2.16: Representación de un vector V en un sistema de coordenadas tridimensional y sus proyecciones en ejes X, Y, Z [86]

La aceleración lineal es la rapidez de cambio de la velocidad lineal, mientras que la velocidad es la rapidez de cambio de la posición lineal. Entonces, la aceleración lineal se define como la segunda derivada de la posición con respecto al tiempo t , y la velocidad lineal, como su primera derivada:

$$\vec{a} = \frac{\partial \vec{v}}{\partial t} = \frac{\partial}{\partial t} \left(\frac{\partial \vec{r}}{\partial t} \right) = \frac{\partial^2 \vec{r}}{\partial^2 t}$$

siendo \vec{a} el vector aceleración lineal, \vec{v} el vector velocidad lineal y \vec{r} el vector posición lineal.

De la misma manera, la rapidez de cambio de la posición angular se denomina como velocidad angular, y se define en la siguiente forma:

$$\vec{\omega} = \frac{\partial \vec{\theta}}{\partial t}$$

siendo $\vec{\omega}$ el vector velocidad angular y $\vec{\theta}$ el vector posición angular [87].

Entonces, la aceleración lineal se mide en metro por segundo al cuadrado (m/s^2), mientras que la velocidad angular se expresa en radián o grado por segundo (rad/s ó deg/s).

2.4.2. Ángulos de Euler

Un sistema de referencia ortogonal puede definirse en base a su orientación con respecto a otro sistema. Los ángulos de Euler son una de las representaciones de orientación más utilizadas en las IMU, donde la orientación se expresa con tres ángulos RPY en el espacio: *roll* (alabeo), *pitch* (cabeceo), y *yaw* (guiñada). Estos ángulos representan las tres rotaciones sobre los ejes X, Y, y Z respectivamente, y se denotan con los símbolos γ , β , y α en la Figura 2.17.

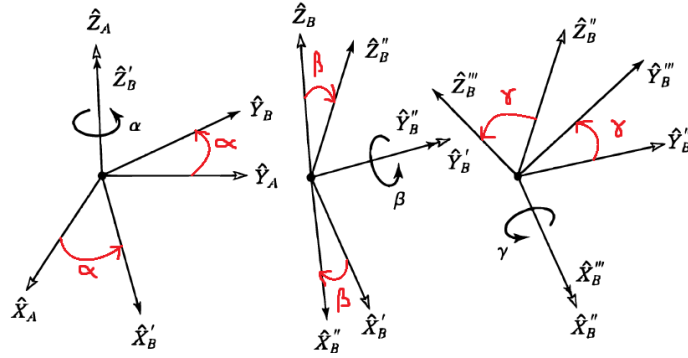


Figura 2.17: Serie de tres rotaciones sobre los ejes del sistema solidario al cuerpo A hasta llegar al sistema resultante B'' [88]

En la Figura 2.17, se puede ver cómo la orientación de un objeto se define por una serie de tres rotaciones que se efectúan alrededor de los ejes del sistema solidario al cuerpo, llamado **A**. Primero se rota un ángulo α alrededor de \hat{Z}_A , luego se gira un ángulo β alrededor de $\hat{Y}_B = \hat{Y}'_B$, resultante del primer giro, y posteriormente se rota un ángulo γ alrededor del eje \hat{X}''_B . La representación de Euler en este orden de rotaciones se denomina como la secuencia ZYX, ya que primero se rota alrededor del eje Z, luego del eje Y, y por último se gira sobre el eje X. Entonces, siendo A el sistema de referencia al cuerpo, el cuerpo tendrá el sistema de referencia final B'' luego de realizar el cambio en orientación [89].

3. Materiales y métodos

3.1. Hardware

3.1.1. Dispositivo de realidad virtual

El modelo del dispositivo de RV que fue utilizado para el proyecto es Reverb G2, desarrollado por HP, una empresa estadounidense con sede en Palo Alto, California, en colaboración con Microsoft y Valve Corporation (Figura 3.1). Lanzado a finales de 2020, el modelo HP Reverb G2 incluye unas gafas con alta resolución visual de 2160×2160 píxeles por ojo, y con una frecuencia de actualización de 90 Hz [90]. Estas características contribuyen a una calidad visual estable y a una experiencia más fluida, lo cual disminuye las náuseas y la fatiga ocular, que son los típicos síntomas de *cybersickness*. Además, ofrecen un seguimiento de seis grados de libertad para que el jugador pueda moverse libremente en el espacio virtual. Aunque las gafas deben estar conectadas a la toma de corriente, el cable tiene una longitud de 6 metros, por lo que el límite máximo de zona de movilidad del jugador es 56,5 metros cuadrados (m^2) aproximadamente, tomando la mitad del área del círculo con radio de 6 metros. Teniendo en cuenta de que la medida estándar para un living se ubica entre los 15 y 20 m^2 [91], el cable no resulta ser una limitación al rango de movimiento de la persona. Por otro lado, las gafas están recubiertas por almohadillas faciales reemplazables, lo cual permite una mayor comodidad y mejor ajuste a la cabeza, disminuyendo la probabilidad de sensación de cinetosis.



Figura 3.1: Casco (izquierda) y los controladores (derecha) de HP Reverb G2 [92]

Teniendo en cuenta el rango de edad predominante en los pacientes con EP, sería preferible que no tuvieran que utilizar controladores de mano y que los dispositivos de realidad virtual incorporaran el seguimiento de las manos (*hand tracking*). En la fase de testeo de este proyecto, se utiliza el dispositivo HP Reverb G2 que no tiene la función de *hand tracking*. Sin embargo, los controladores de HP Reverb G2 presentan un diseño ergonómico y más compacto en comparación con sus modelos anteriores, lo que permite al usuario una interacción intuitiva. De todas maneras, la aplicación *BrainWalkVR* fue programada con la plataforma de *SteamVR*, lo que garantiza su compatibilidad y ejecución fluida en diversos modelos de dispositivos de RV. Entre ellos se incluyen el HTC Vive, Meta Quest 2, Meta Quest 3, Meta Quest Pro, y PICO, permitiendo a los usuarios

disfrutar de la experiencia sin inconvenientes en una variedad de dispositivos.



Figura 3.2: Uso de los dispositivos de RV [93]

3.1.2. Modo de uso

Para usar la aplicación, el usuario debe colocarse el casco en su cabeza y sostener los dos controladores en ambas manos. Cada controlador tiene: dos botones de letras (Izquierdo: X, Y ; Derecho: A, B); el botón de Windows; el botón de menú; un *thumbstick* (joystick manejada con pulgar); un botón de *Grab*, y un botón de *Trigger* (Figura 3.3) A continuación, se describen las funcionalidades de cada parte del controlador en este proyecto, *BrainWalkVR*.

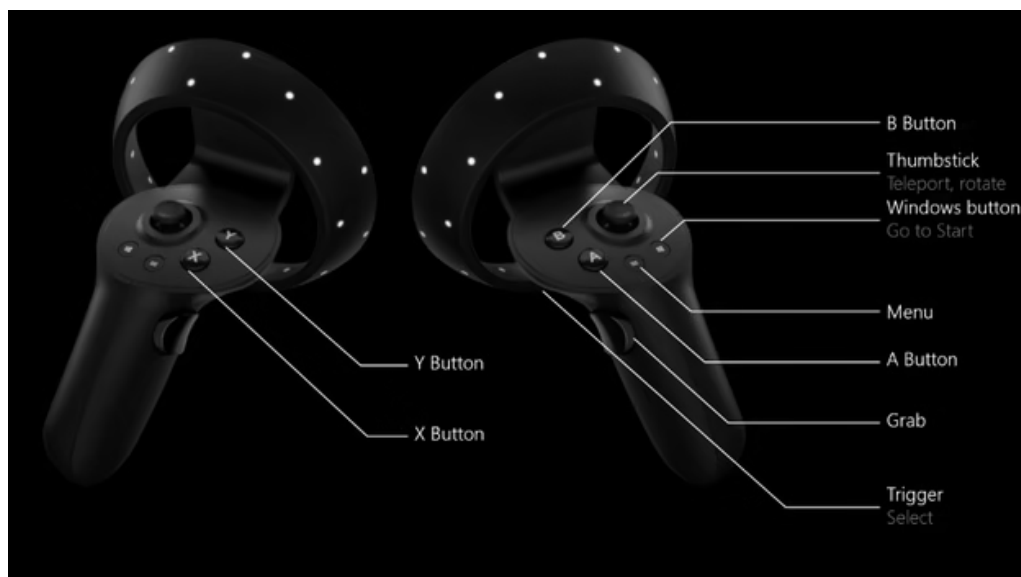


Figura 3.3: Partes del controlador de HP Reverb G2 [94]

- **Botón *Trigger* o *Select*:** se aprieta con el dedo índice. En las escenas de inicio y de finalización, aparecerá un puntero láser en ambas manos virtuales para que el usuario pueda apuntar los elementos de interfaz gráfica en los menús. En la escena principal, cuando el usuario toque el botón de menú para abrir la interfaz de usuario, también aparecerán los punteros láser.
- **Botón *Grab*:** se aprieta con el dedo medio. El usuario debe acercarse a un objeto lo suficiente para que el color del objeto se cambie indicando que ha sido seleccionado. Luego, si aprieta el botón de *Grab* y lo mantiene apretado, el objeto se sostendrá en la mano del usuario. Cuando deje de apretar el botón, se soltará el objeto agarrado.
- **Botón *Menu*:** se toca con el pulgar. Al tocar este botón del controlador izquierdo, se abre una interfaz gráfica durante la sesión, donde el usuario puede regular el volumen de música de fondo, volver a la escena de inicio o finalizar la sesión actual.

En la aplicación, el usuario podrá realizar todas las acciones con sólo usar estos tres botones, ya que el usuario se desplaza y gira físicamente sin usar el *thumbstick*. De esta manera, se logra que el sistema sea simple y *user-friendly*, teniendo en cuenta que la mayoría de pacientes con Parkinson tiene edad promedio mayor a 50 años, por lo que es probable que tengan poca experiencia con los dispositivos de RV.

3.2. Software

3.2.1. *Unity*

El software de la aplicación *BrainWalkVR* fue diseñado con *Unity*, que es un motor de desarrollo en tiempo real que se utiliza para crear experiencias interactivas en 2D y 3D. Lanzada por la compañía *Unity Technologies* en 2005, esta multiplataforma permitió la creación de diversos videojuegos y realidad extendida, que sean compatibles en PC, videoconsolas y móviles. Para trabajar con *Unity*, el usuario debe descargar el *Unity Hub* y el *Unity Editor*: el primero es una aplicación con la que uno puede acceder al ecosistema de *Unity* para administrar sus proyectos e instalar distintas versiones del Editor, mientras que el *Unity Editor* es la plataforma por la cual el usuario crea sus aplicaciones y juegos utilizando las herramientas que ofrece el Editor [95]. El Editor de *Unity* es compatible con los siguientes tres Entorno de Desarrollo Integrado (IDE), que son las aplicaciones de software que facilitan a los programadores desarrollar código de software: *Visual Studio*, *Visual Studio Code* y *JetBrains Rider* [96]. Para el presente trabajo, se eligió trabajar con el lenguaje de programación C# en *Visual Studio*.

Unity es uno de los programas de software más utilizados para la creación de experiencias de RV en la actualidad, junto a *Unreal Engine* [97]. Para este proyecto, se eligió trabajar con *Unity* por su facilidad de aprendizaje y amplia compatibilidad con los dispositivos [98]. Si bien actualmente la mayor utilidad de *Unity* es el desarrollo de videojuegos, esta multiplataforma está emergiendo como una herramienta potencial para aplicaciones en ámbito de salud. Un ejemplo de los proyectos relacionados con la medicina es TOMMI, una experiencia de RV diseñada por *Softcare Studios* para ayudar a los pacientes pediátricos a enfrentar y adaptarse mejor a los tratamientos e internación en el hospital [99]. Otro ejemplo es ORamaVR que consiste en un simulador de RV para procedimientos médicos como la artroplastia de rodilla, donde los estudiantes y profesionales

de salud puedan aprender y practicar sus técnicas en un entorno realista, sin poner en riesgo a pacientes reales.

Con las herramientas que ofrece el Editor de Unity, el desarrollador puede crear y editar escenas y objetos, programar sus comportamientos, y por ende crear aplicaciones interactivas tanto en segunda como en tercera dimensión (3D). El Editor está compuesto por las siguientes ventanas principales que le permite al usuario desarrollar su proyecto de manera dinámica y eficiente: la ventana del proyecto (*Project Window*), de escena (*Scene View*), de jerarquía (*Hierarchy Window*), del inspector (*Inspector Window*) y del juego (*Game View*) (Figura 3.4). La versión del Editor usada en el presente trabajo es 2021.3.16f1 Long Term Support (LTS).

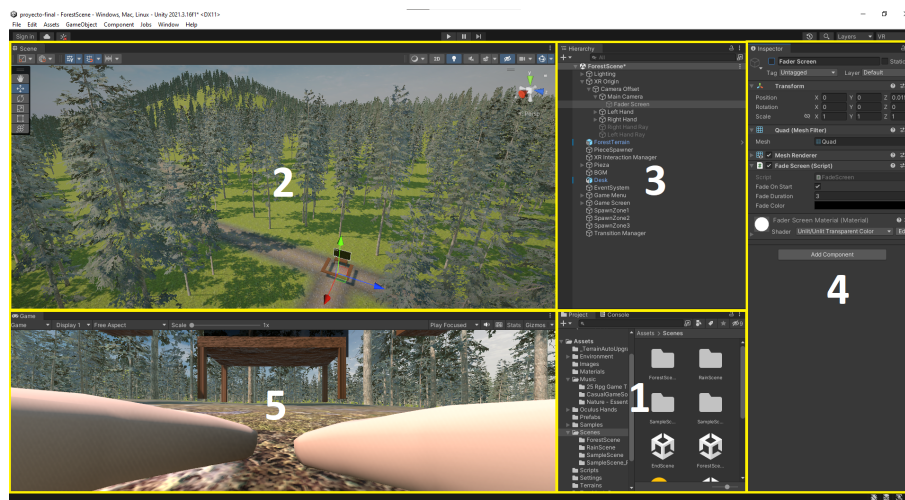


Figura 3.4: Las ventanas principales del Unity Editor. 1. *Project Window* 2. *Scene View* 3. *Hierarchy Window* 4. *Inspector Window* 5. *Game View*

En la ventana *Project*, el usuario puede acceder y administrar los *assets* de su proyecto. Un *asset* es una representación de cualquier elemento que puede ser utilizado en su juego o proyecto, tales como un modelo 3D, una imagen, un archivo de sonido, entre otros archivos compatibles con *Unity* [100].

La vista *Scene* es una ventana interactiva donde el usuario puede seleccionar y posicionar los *GameObjects* rápidamente. Cuando un *asset* se usa en una escena, se convierte en un *GameObject*, que puede representar un personaje, un objeto, un escenario o la cámara. Estos *GameObjects* de la escena actual se ordenan en jerarquía en la ventana *Hierarchy*: por defecto, los *GameObjects* se mostrarán en el orden en que se fabrican, y el usuario puede desplazarlos estableciendo relaciones de hijos o padres. Cuando un *GameObject* es el hijo de otro, este heredará el movimiento y rotación de su *GameObject* padre. Esta funcionalidad es útil en el momento en que el usuario quiera desarrollar un *GameObject* que tenga integrados varios *GameObjects*.

Cada *GameObject* contiene *Components*, que son las piezas funcionales que determinan el comportamiento y propiedades de ese objeto en la escena actual. Los componentes se muestran en la ventana *Inspector*, donde el usuario puede agregar, eliminar o editar las propiedades del

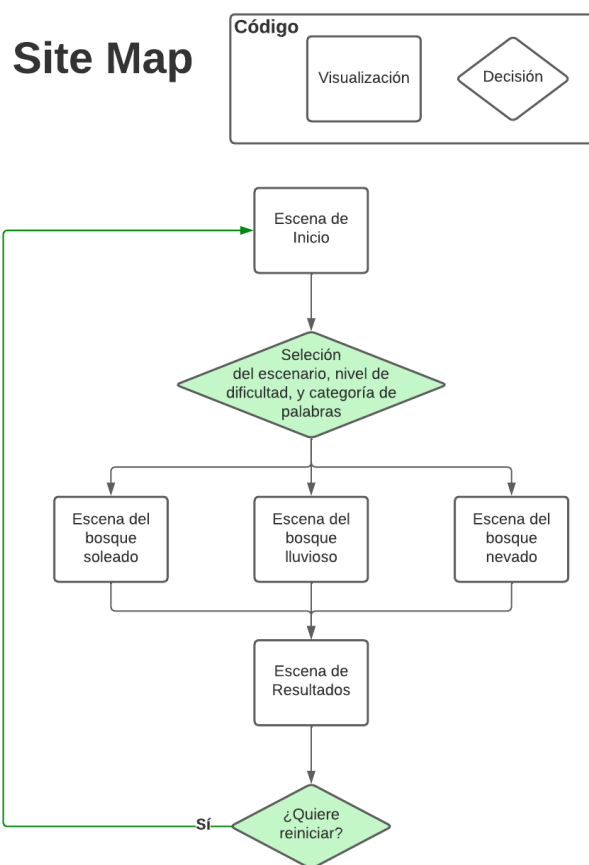
GameObject. A continuación, se enumeran los principales componentes que se utilizaron en el presente trabajo [101]:

- **Transform:** define la posición, rotación, y escala del *GameObject* en la escena, en el sistema de coordenadas X, Y, Z.
- **Audio Source:** reproduce un *audio clip* en la escena.
- **Script:** es un archivo de código de software que determina el comportamiento del *GameObject*, ejecutando funciones. Se compila en uno de los IDE compatibles con *Unity*, y se escribe en lenguajes de programación C#.
- **Rigidbody:** le otorga el comportamiento físico al *GameObject*: por ejemplo, el objeto puede tener cierta masa en kilogramos, y responder a la gravedad.
- **Collider:** define la forma de un *GameObject* para los propósitos de colisiones físicas.

Por otro lado, en este proyecto se utilizó el paquete de *XR Interaction Toolkit*, una herramienta desarrollada por la empresa fundadora que permite crear un sistema interactivo en las experiencias de RA y RV. Gracias a este paquete, el desarrollador no tiene que crear las interacciones desde cero, sino que simplemente puede agregar los componentes a los *GameObjects* para darles interactividad.

3.2.2. Flujo del Programa

El software del proyecto se divide en tres bloques principales: la escena de inicio, donde el usuario podrá seleccionar el nivel de dificultad del entrenamiento y el tipo de escenario; la escena principal, que se basará en el escenario elegido previamente por el usuario y es donde efectivamente el jugador realizará ejercicios de entrenamiento y podrá ver la cantidad de errores cometidos en la sesión, y por último, la escena final, donde el jugador puede decidir si finalizar la sesión, o comenzar una nueva. Este flujo del sistema se muestra a través del diagrama de flujo en la Figura 3.5.

Figura 3.5: *Site map* del software

En cada sesión de entrenamiento, el usuario debe completar una serie de 6 palabras que pertenecen a una determinada categoría. Estas palabras se mostrarán una por una en orden sobre una pantalla principal, a medida que el jugador va completando la palabra con las letras que faltan. Para completar una letra de la palabra, el usuario debe primero seleccionar y agarrar una de las piezas con grabado de letras, que estarán distribuidas a su alrededor sobre los paneles de madera. Luego, el jugador debe llevar la pieza seleccionada y colocarla sobre la mesa que se encuentra en frente de la pantalla principal. Si la pieza contiene a una de las letras faltantes, se completará un espacio de la palabra en la pantalla (marcado con guión bajo). En caso de que la pieza contenga una letra incorrecta, el usuario escuchará un sonido indicando que ha seleccionado una pieza equivocada, y la pieza colocada sobre la mesa desaparecerá.

Por ejemplo, la palabra a completar es «E S P I N A C A», y en la pantalla principal aparece como «E _ _ _ A _ A». Si el jugador coloca una pieza con la letra «S» sobre la mesa, la pantalla ahora mostrará «E S _ _ A _ A». Sin embargo, si el jugador coloca una pieza con la letra «Z», la pantalla no se modificará y la pieza colocada desaparecerá por ser incorrecta. Cuando el jugador termina de completar una palabra, la pantalla mostrará la siguiente palabra de la misma categoría, y así sucesivamente hasta que se completen todas las 6 palabras de la serie. Todas las acciones del

usuario y los cambios en las escenas se resumen en el flujo de usuario (*User Map*) en la Figura 3.6, donde se muestra la ruta que debe seguir el usuario para terminar una sesión de entrenamiento en la aplicación. Tanto el *site map* como el *user map* fueron diseñados en la página de Lucidchart [102].

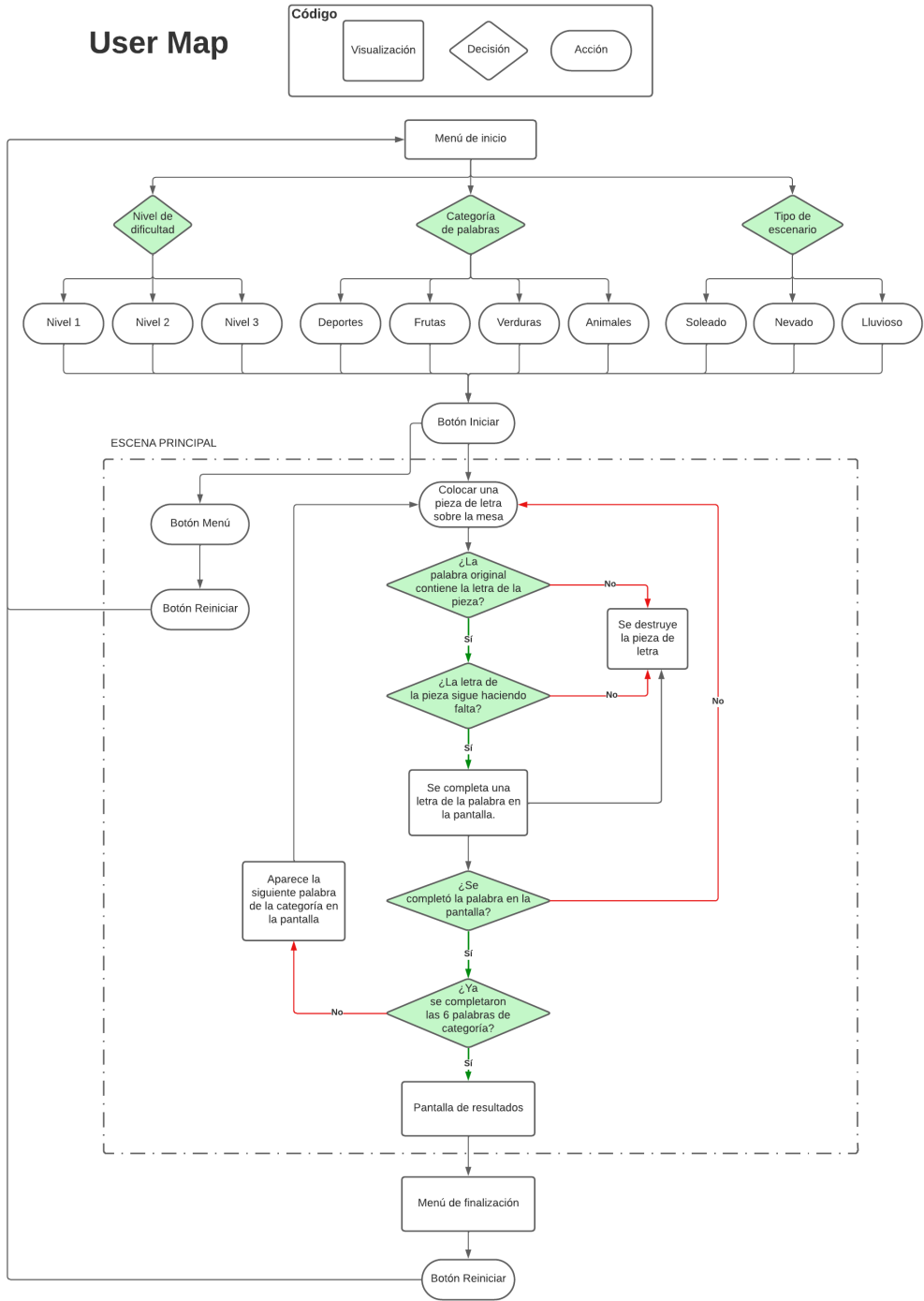


Figura 3.6: Flujo de usuario del software

3.2.3. Escena de Inicio

Al iniciar la aplicación, el jugador se encontrará en medio de un escenario de bosque soleado (Figura 3.7) Se decidió elegir este tipo de ambiente en base a un estudio de investigación, donde se analizaron los impactos del tipo de paisaje sobre la ansiedad, depresión y estrés de los pacientes, usando un simulador de RV [103]. Como se mencionó anteriormente en la sección 2.2.1, la mitad de los pacientes con EP experimentan síntomas depresivos, que conducen a una disminución de calidad de vida y una aceleración de deterioro tanto cognitivo como motor. En dicha investigación, se demostró que los entornos virtuales con mayor efecto restaurador para los pacientes fueron la pradera y el bosque, que se caracterizan por tener mayor apertura, más elementos verdes, un cielo más azul y mayor exposición a la luz solar. Además, los ambientes con colores fríos resultaron más relajantes para los pacientes a la hora de evaluar sus efectos restauradores.

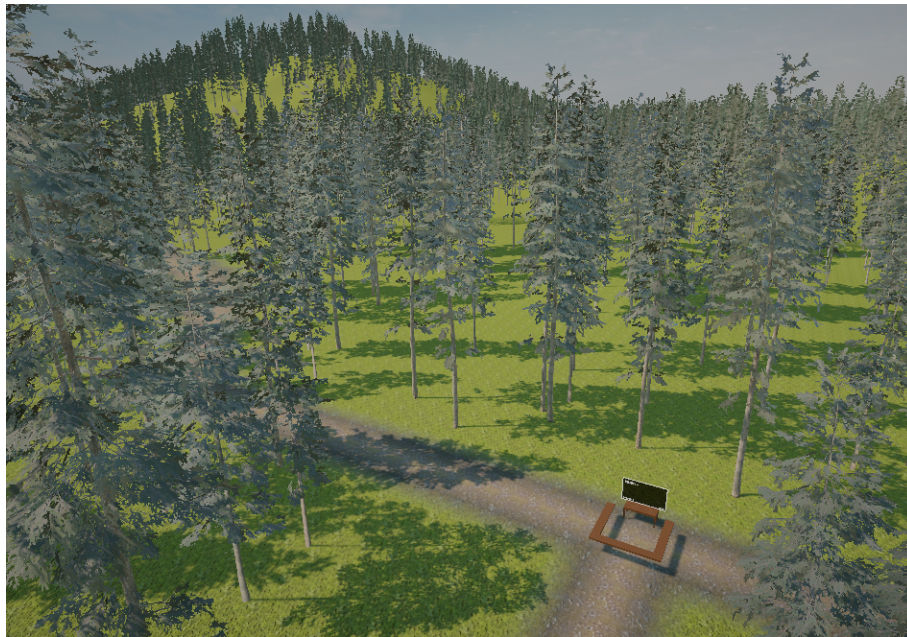


Figura 3.7: Escena del bosque soleado

Por esta razón, se eligió diseñar un bosque como el terreno principal. Los componentes del ambiente, tales como los árboles, las diferentes texturas del terreno y el cielo, fueron importados de los assets sin cargo publicados en *Unity Asset Store* [104, 105, 106, 107, 108]. La iluminación fue diseñada con una temperatura de luz entre 5000 y 6500 Kelvin, para imitar el color blanco frío de la luz del día [109].

Por otro lado, en la escena de inicio aparecerá un menú de inicio en frente del jugador al iniciar la aplicación (Figura 3.8). En este menú, el usuario podrá seleccionar los tres parámetros: el tipo de escenario, la categoría de palabras utilizadas en la sesión actual, y el nivel de dificultad del entrenamiento.



Figura 3.8: Menú de selección en la escena de inicio

Para que el usuario pueda elegir un escenario a su gusto, se le ofrecen tres tipos de ambientes diferentes: bosque soleado, bosque nevado y bosque lluvioso (Ver Figuras 3.7, 3.9, 3.10). De esta manera, el paciente podrá realizar la actividad en un ambiente a su preferencia, lo cual aumentará sus motivaciones para rehabilitación.

La categoría de palabras puede ser deportes, frutas, verduras o animales, y cada categoría corresponde a una lista secuencial de 6 palabras en letra capital, que estará guardada en los scripts. Se eligieron las palabras más usadas en la vida cotidiana, ya que los pacientes con Parkinson presentan dificultad para recordar palabras conocidas. Las palabras de cada categoría son las siguientes:

Categoría	Palabras (en mayúsculas)
Deportes	FUTBOL, TENIS, HOCKEY, NATACION, BOXEO, PINGPONG
Frutas	MANZANA, BANANA, NARANJA, MANDARINA, SANDIA, FRUTILLA
Verduras	ZANAHORIA, ESPINACA, PEPINO, TOMATE, BERENJENA, LECHUGA
Animales	ELEFANTE, CABALLO, GALLINA, CONEJO, PERRO, CAMELLO

Cuadro 1: Lista de palabras por categoría

Además, el jugador podrá elegir el nivel de dificultad entre 1, 2, y 3. El grado de dificultad se refiere a la cantidad de letras reveladas en cada palabra: si el nivel de dificultad es 1, habrá 3 letras ya completas y el jugador deberá completar el resto; si es 2, habrá 2 letras ya completas, y

si es nivel 3, habrá solo una letra revelada, por lo que será más difícil para el jugador adivinar y completar las palabras.

En el menú de inicio, hay dos columnas de checkbox: una de lado izquierdo, para seleccionar la categoría de palabras, y otra de lado derecho, para elegir el tipo de escenario. El usuario podrá elegir sólo una opción en cada columna de checkbox. Por último, el grado de dificultad se selecciona por un slider, donde el usuario puede elegir una de las tres opciones deslizando el cursor circular. Luego de definir todos los parámetros, el jugador debe presionar el botón de «INICIAR» para comenzar la sesión de entrenamiento.

3.2.4. Escena principal

En la escena principal, el usuario realizará ejercicios de entrenamiento en el bosque con uno de los tres climas: soleado, nevado y lluvioso (Figuras 3.7, 3.9, 3.10). Al iniciar la escena, el jugador escuchará una música de fondo en común, importada de *Unity Asset Store* [110]. Luego, según el escenario habrá efecto de sonido adicional que caracteriza al clima del ambiente: por ejemplo, en el bosque nevado habrá sonido del viento leve, mientras que en el bosque lluvioso, habrá sonido de la lluvia. Todos estos archivos de audio también fueron importados de *Unity Asset Store* sin cargo [111].

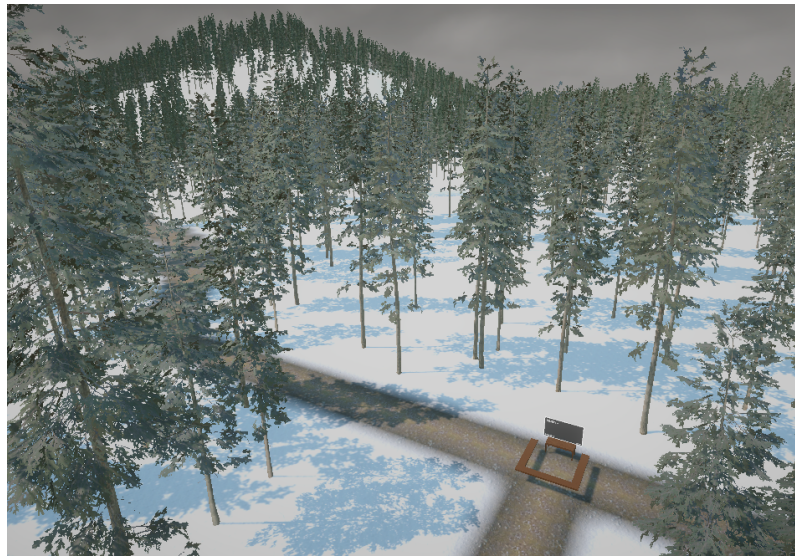


Figura 3.9: Escena del bosque nevado

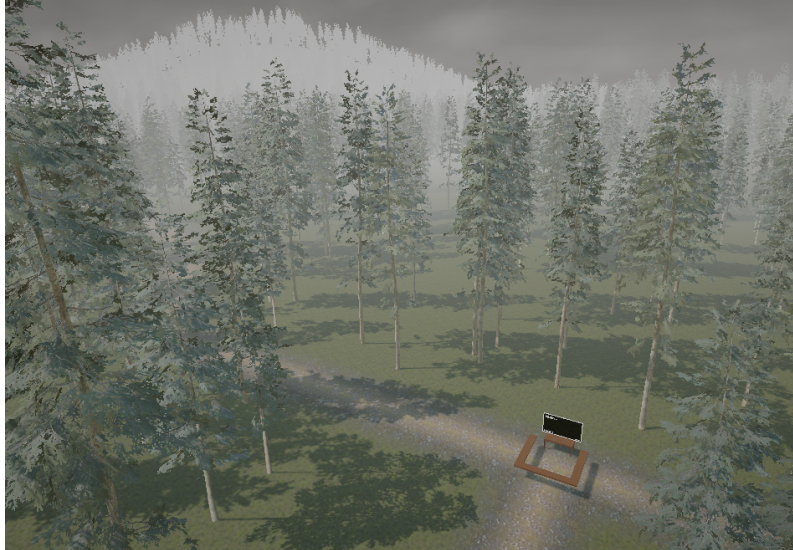


Figura 3.10: Escena del bosque lluvioso

El espacio de entrenamiento está compuesto por una pantalla principal, una mesa en frente de la pantalla, y sus bordes están delimitados por paneles de madera (Figura 3.11) En la pantalla, aparecerán la categoría de palabras seleccionada en el menú de inicio, la palabra a completar, la cantidad de errores que comienza en valor nulo, y el número de ronda que empieza en 1 y va incrementando a medida que el jugador va completando las palabras. La última ronda sería el número 6, porque para cada categoría hay una lista de 6 palabras.



Figura 3.11: Espacio de entrenamiento



Figura 3.12: Pantalla principal

En cada sesión de entrenamiento, el usuario debe completar una serie de 6 palabras que pertenecen a una determinada categoría. Estas palabras se mostrarán una por una en orden sobre la pantalla principal, a medida que el jugador va completando la palabra con las letras que faltan. Por ejemplo, en la Figura 3.11, la palabra a completar es «PERRO», y en la pantalla aparece «_E_RO», teniendo tres letras reveladas.

Para completar una letra de la palabra, el usuario debe primero seleccionar y agarrar una de las piezas con grabado de letras, que estarán distribuidas a su alrededor sobre los paneles de madera como se ve en la Figura 3.11. Estas piezas son cubos con textura y brillo metálico, de color dorado, para que se distingan bien de su alrededor (Figura 3.13). Al acercarse y tocar una pieza, el cubo cambiará de color indicando que ha sido seleccionada (Figura 3.14), y al agarrarlo, el jugador escuchará un audio de feedback indicando que ha sujetado la pieza [112], y el cubo se ubicará en la mano del jugador de manera tal que pueda ver claramente la letra de la pieza seleccionada (Figura 3.15).

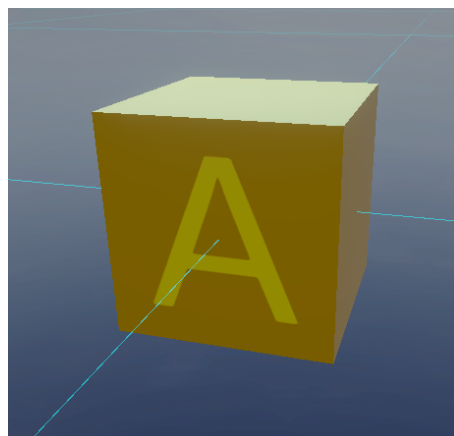


Figura 3.13: Prefab de la pieza de letra

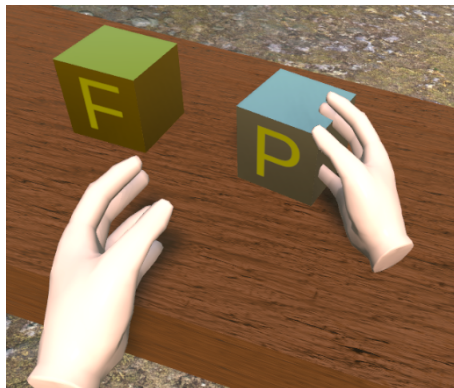


Figura 3.14: Selección de una pieza



Figura 3.15: Posición de una pieza agarrada en la mano

Luego, el jugador debe llevar la pieza seleccionada y colocarla sobre la mesa que se encuentra en frente de la pantalla principal. Si la pieza contiene a una de las letras faltantes, se completará un espacio de la palabra en la pantalla (marcado con guión bajo). En caso de que la pieza contenga una letra incorrecta, el usuario escuchará un sonido indicando que ha seleccionado una pieza equivocada, y la pieza colocada sobre la mesa desaparecerá. Siguiendo el ejemplo anterior, la palabra a completar es «P E R R O», y en la pantalla principal aparece como «_ E _ R O». Si el jugador coloca una pieza con la letra «P» sobre la mesa, la pantalla ahora mostrará «P E _ R O». Sin embargo, si el jugador coloca una pieza con la letra «Z», la pantalla no se modificará y la pieza colocada desaparecerá por ser incorrecta y se le sumará un 1 en la cantidad de errores cometidos. Cuando el jugador termina de completar una palabra, se pasará a la siguiente ronda y la pantalla mostrará la palabra que le sigue en la lista, y así sucesivamente hasta que se completen todas las 6 palabras de la categoría.

Por otro lado, durante el entrenamiento el usuario puede apretar el botón de menú en el controlador izquierdo para abrir una interfaz de usuario (Figura 3.16) En el menú, hay dos sliders con los que el jugador puede regular el volumen de música de fondo y el de sonidos de interacción, respectivamente. Los sonidos de interacción serían por ejemplo el audio que se produce cuando se coloca una pieza incorrecta sobre la mesa, o el sonido de victoria cuando el jugador finaliza una sesión de entrenamiento. Además, el usuario puede elegir reiniciar la sesión volviendo a la escena

de inicio, o finalizar la sesión yendo a la escena final. De esta manera, el jugador tiene más libertad para comenzar y terminar el entrenamiento cuando sea necesario.



Figura 3.16: Interfaz de usuario

3.2.5. Escena Final

En la escena final, el usuario de nuevo se encontrará en medio del escenario de bosque soleado, de la misma manera que en la escena de inicio, y estará el menú de finalización al frente del jugador (Figura 3.17). Acá, simplemente se puede elegir si terminar la sesión de entrenamiento, o iniciar una nueva sesión de entrenamiento. Si se presiona el botón «SALIR», se da por finalizada la aplicación, y si se aprieta el botón «REINICIAR», volverá a aparecer la escena de inicio donde el jugador elige el ambiente, la categoría de palabras y el nivel de dificultad para una sesión nueva.



Figura 3.17: Menú de finalización en la escena final

3.3. Fase de testeo

3.3.1. Protocolo

Para validar la utilidad de la aplicación, se realizaron pruebas de manera presencial en el laboratorio de Ciencia de la Vida, en la Sede Distrito Tecnológico del Instituto Tecnológico de Buenos Aires (ITBA). Las pruebas se realizaron en un espacio $3 \times 3 \text{ m}^2$ sin obstáculos, para que los voluntarios puedan moverse con el casco sin el riesgo de tropezarse. La dimensión del espacio fue previamente marcada con cinta amarilla como se muestra en la Figura 3.18.



Figura 3.18: Espacio $3 \times 3m^2$ donde se realizaron las pruebas

Las pruebas se realizaron con 11 voluntarios (2 hombres y 9 mujeres) de población sana con edad media de $51 \pm 3,1$ años, sin experiencia previa con la realidad virtual. Si bien en mayoría los primeros síntomas de la enfermedad aparecen a los 60 años, hay un 10% de casos en que los síntomas inician antes de los 50 años. Por esta razón, se decidió tomar un grupo de personas cuya edad esté en el rango de aparición de primeros síntomas de EP [113, 114]. El objetivo de esta fase de testeo es evaluar la usabilidad del sistema y el grado de *cybersickness* que produce la aplicación, y medir los parámetros de movilidad para validar el potencial de la aplicación como futura herramienta de rehabilitación de marcha. Por ende, se realizaron las pruebas con personas sanas, en el lugar de pacientes reales.

Para la selección de sujetos de estudio, las personas tuvieron que completar un formulario de Google Forms donde se recolectó información sobre los datos personales, la disponibilidad horaria, y los antecedentes de salud. En cuanto al historial de salud, se preguntó sobre el uso de anteojos para la comodidad de la persona al colocarse el casco de RV, y sobre si padece alguna enfermedad neurológica, trastorno cognitivo o motor. Además, se preguntó si la persona habitualmente sufre de mareos, náuseas vértigo o fatiga visual, para poder evaluar efectivamente los síntomas de *cybersickness* producidos por la aplicación. Por último, se tomó en cuenta si la persona tiene experiencia previa en RV para ver cuán difícil resulta aprender a usar la aplicación, sin tener conocimientos previos sobre dicha tecnología.

Luego de seleccionar los sujetos de estudio, se realizaron las pruebas siguiendo el protocolo de evaluación previamente diseñado, donde se detallan los procedimientos de la prueba con sus duraciones estimadas (Tabla 2). Cada participante tuvo que firmar un consentimiento para estar informado sobre los objetivos del proyecto, qué datos se toman en la fase de testeo, y sobre la posibilidad de que tenga síntomas relacionados al *cybersickness*. El consentimiento informado se adjunta en el Anexo (Figura 9.1).

N°	Duración estimada (min)	Pasos a realizar
1	5	La persona debe firmar el consentimiento. Se ajusta la altura del piso del dispositivo para la persona.
2	5	Se da explicación sobre el uso de dispositivo en la aplicación: funcionalidad de botones, flujo del programa.
3	2	Se le coloca el sensor Xsens DOT para obtener los parámetros de marcha en MVN SDK. Se le pega con la cinta en el pie derecho.
4	3	Se le coloca el casco de realidad virtual al voluntario. Se otorga un tiempo de adaptación al entorno virtual.
5	10	La persona realiza una sesión de entrenamiento: tendrá libertad en elegir el escenario. La categoría de palabras y el nivel de dificultad para la prueba serán: Frutas, Nivel 2. Se toman los datos relacionados a la marcha con Xsens. Se registra la cantidad de errores durante la sesión.
6	5	La persona debe completar el CSQ-VR y el SUS en papel. Mientras, la persona del siguiente turno realiza el paso 1.

Cuadro 2: Procedimientos de prueba con sus duraciones estimadas

Como se mencionó anteriormente, en esta fase de testeo se evaluarán tres aspectos en la aplicación: la usabilidad del sistema, el grado de *cybersickness* producido, y el grado de movilidad que induce el entrenamiento. Para medir los primeros dos aspectos, se utilizaron los cuestionarios Sistema de Escala de Usabilidad (SUS) y Cuestionario de Cíbermareo en Realidad Virtual (CSQ-VR), que fueron traducidos en español previo a la fase de testeo para que los voluntarios puedan completar. Los cuestionarios originales en inglés se adjuntan en el Anexo (Figuras 9.2, 9.5, 9.6). El grado de movilidad se midió con el sensor Xsens DOT, cuyos datos luego fueron procesados en el código de Python para obtener la distancia recorrida total, la cantidad de pasos y la cantidad de giros.

3.3.2. Sistema de Escala de Usabilidad (SUS)

El Sistema de Escala de Usabilidad (SUS) es un cuestionario basado en la escala de Likert de cinco puntos, que mide la usabilidad de un software. El SUS es uno de los métodos de medición de usabilidad más utilizados en la experiencia de usuario, porque mide no solamente la usabilidad de un sistema, sino la habilidad del usuario de aprender a usar el sistema. Además, sus resultados siguen siendo confiables aunque se trabaje con muestras pequeñas [115]. Entonces, este método de evaluación resulta beneficioso para este proyecto ya que permite medir la facilidad del usuario para aprender y adaptarse al sistema con un número limitado de participantes.

El cuestionario de SUS está compuesto por diez preguntas en total, cuyos puntajes se definen por la posición de escala. Tiene cinco preguntas con enfoque positivo (ítems 1, 3, 5, 7, 9) y cinco preguntas con enfoque negativo (ítems 2, 4, 6, 8, 10), que se encuentran intercaladas para evaluar la usabilidad del sistema (Anexo: Figuras 9.3 y 9.4). Para normalizar los puntajes de cada pregunta, se resta 1 al puntaje de una pregunta con enfoque positivo, mientras que para las preguntas con enfoque negativo, se resta el puntaje a 5. De esta manera, los puntajes individuales quedan en

el rango entre 0 y 4. Luego, se suman todos los puntajes y se multiplica por 2,5 para obtener finalmente el puntaje final dentro del rango del 0 al 100 [116]. Este resultado del SUS no se trata de un porcentaje, sino que se debe buscar a qué rango percentil pertenece para interpretar finalmente el nivel de usabilidad del sistema.

La interpretación del resultado se basa en la curva de SUS, donde se muestran los rangos percentiles en función de los puntajes de SUS, que se ha obtenido a partir de una base de datos obtenidos en los 500 estudios de investigación que se realizaron a más de 5.000 usuarios (Figura 3.19). En este gráfico, se puede saber a qué rango percentil pertenece un puntaje SUS en bruto, permitiendo así determinar qué tan bueno es el puntaje SUS en comparación con la base de datos. Según el gráfico, se puede considerar que un puntaje mayor o igual a 68 puntos indica que se encuentra arriba del valor promedio, ya que es donde corresponde el percentil 50.

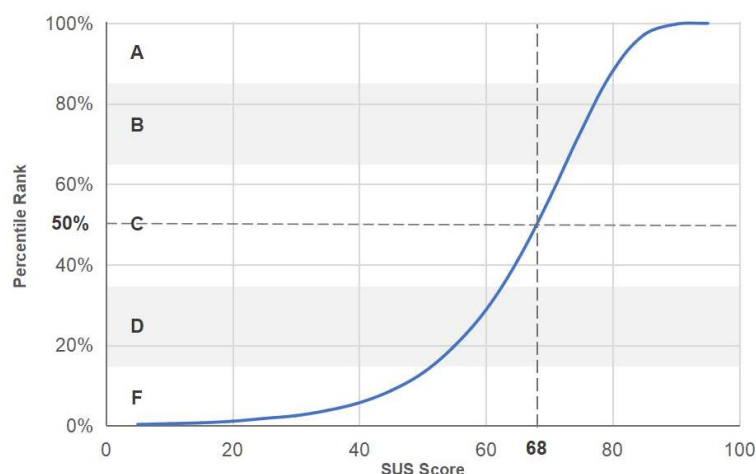


Figura 3.19: Curva de SUS con los rangos percentiles en función de los puntajes en bruto [117]

Los rangos percentiles de SUS están asociados con cuatro tipos de escalas diferentes, que se pueden utilizar para evaluar cualitativamente la usabilidad del sistema (Figura 3.20):

- **Calificaciones (*grade*):** van de la A a la F, en orden decreciente de la usabilidad.
- **Adjetivos:** la escala contiene «Mejor imaginable», «Excelente», «Bueno», «Regular», «Malo», «Peor imaginable», en orden decreciente de la usabilidad. El rango entre «Mejor imaginable» y «Regular» se considera aceptable.
- **Aceptabilidad:** la escala contiene «Aceptable», «Marginalmente aceptable», «No aceptable».
- **Net Promoter Score (NPS):** es un indicador para determinar la lealtad de los clientes a una empresa en los estudios de experiencia del cliente. Al relacionar NPS con SUS, resultó que los puntajes SUS mayores a 81 son promotores, es decir que son personas leales y entusiastas; los puntajes entre 70 y 81 son pasivos, que están satisfechos con el servicio pero no lo suficiente como para considerarse promotores, y el resto serán detractores que son insatisfechos y podrían incluso desalentar a otros de no volver a usar el servicio [118].

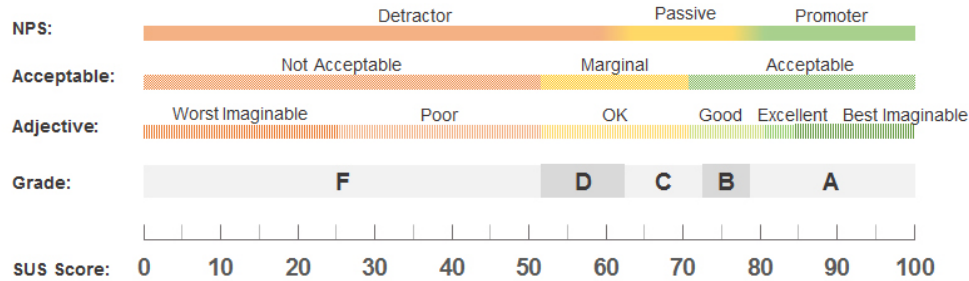


Figura 3.20: Calificaciones con letras, escala de adjetivos, aceptabilidad y Net Promoter Score asociados con los puntajes del SUS en bruto

En el presente trabajo, cada voluntario completará el cuestionario de SUS después de realizar una sesión de entrenamiento con la aplicación *BrainWalkVR*, para evaluar la usabilidad y la habilidad de aprender a usar el sistema.

3.3.3. Cuestionario de Cibermareo en Realidad Virtual (CSQ-VR)

Como se explicó en la sección 2.3.2, el *cybersickness* o cibermareo es un conjunto de efectos indeseados como mareos o náuseas, que puede presentar una persona durante o después de su exposición a un entorno virtual. El Cuestionario de Cibermareo en Realidad Virtual (CSQ-VR) es una herramienta derivada del Cuestionario de Neurociencia de Realidad Virtual (VRNQ) que cuantifica los Síntomas y Efectos Inducidos por la Realidad Virtual (VRISE), midiendo así el grado de *cybersickness* que produce una experiencia de RV [119]. El cuestionario utiliza escala de Likert de siete puntos para evaluar los síntomas principales de *cybersickness*: náuseas, desorientación y síntomas oculomotores. Cada una de las tres categorías contiene dos preguntas, y el puntaje total del cuestionario se define por la suma de puntajes de todas las categorías. Entonces, el puntaje total mínimo en el cuestionario es 6 puntos, y el máximo, 42 puntos, mientras que para cada categoría su puntaje mínimo es 2 puntos y el máximo, 14 puntos.

Para el CSQ-VR, no existe un método de interpretación de los puntajes como en el Sistema de Escala de Usabilidad. Por lo tanto, se armó una escala de seis puntos dividiendo el rango entre 6 y 42 puntos en seis rangos. Cada rango indica el grado de *cybersickness* que implica el puntaje que se encuentra en dicho rango (Tabla 3). De esta manera, se pudo interpretar los puntajes totales del CSQ-VR obtenidos en la fase de testeo, para evaluar el grado de *cybersickness* que produce la aplicación de *BrainWalkVR*. Por otro lado, para poder identificar el tipo de síntomas de *cybersickness* que produce mayormente la aplicación, se armó también una escala de seis puntos para los puntajes por categoría, asignando la intensidad de síntomas a cada rango de puntajes, de la misma manera que se hizo para la escala de los puntajes totales (Tabla 4). Así, además de evaluar el grado de *cybersickness* en general, se pudo analizar la intensidad de síntomas relacionados con las náuseas, desorientación y lo oculomotor.

Rango del puntaje total CSQ-VR	Grado de <i>cybersickness</i>
[36 - 42]	Extremo
[30 - 36)	Grave
[24 - 30)	Significativo
[18 - 24)	Moderado
[12 - 18)	Leve
[6 - 12)	Mínimo

Cuadro 3: Tabla de rangos del puntaje total CSQ-VR y el grado de *cybersickness* que corresponde cada rango

Rango del puntaje CSQ-VR por categoría	Intensidad de síntomas
[12 - 14]	Extremo
[10 - 12)	Grave
[8 - 10)	Significativo
[6 - 8)	Moderado
[4 - 6)	Leve
[2 - 4)	Mínimo

Cuadro 4: Tabla de rangos del puntaje CSQ-VR por categoría y la intensidad de síntomas que corresponde cada rango

3.3.4. Sensor inercial Xsens DOT

Para la fase de testeo del proyecto, se eligió trabajar con una Unidad de Medición Inercial (IMU) llamada Xsens DOT, de la empresa *Xsens Technologies B.V.* Este sensor *wearable* permite obtener mediciones de parámetros de movimiento en tiempo real sin restricciones en la movilidad, por lo que se utiliza principalmente en área de rehabilitación, en los estudios biomecánicos en el deporte y en el análisis de la marcha. Actualmente en Fleni, se realizan varios análisis del movimiento con este sensor, lo cual dio pie para usar el dispositivo en el presente proyecto.

Xsens DOT tiene integrados los siguientes sistemas inerciales: acelerómetros, que miden la aceleración lineal; giróscopos, que miden la velocidad angular y proveen la orientación espacial del movimiento, y los magnetómetros, que miden campos magnéticos como referencia para la inclinación. Por otro lado, el procesador integrado del sensor se encarga de muestreo y calibración, también tiene incorporado un algoritmo llamado Xsens Kalman Filter core (XKFCore) para la fusión de sensores (*Sensor Fusion*), que combina la información obtenida por tres sensores inerciales y devuelve resultados pertinentes sobre el movimiento de la persona [120, 121].

Los datos de salida de Xsens DOT están representados en diferentes sistemas de referencia. El sistema de coordenadas local al sensor, llamado S , es fijo al cuerpo del sensor y sus tres ejes se definen por las letras minúsculas x , y , y z (Figura 3.21). Por otro lado, se establece otro sistema de referencia fijo a la Tierra llamado L , conocido como las coordenadas East North Up (ENU), con sus ejes X , Y y Z . En este sistema, el eje X es positivo hacia el este magnético, el eje Y hacia

el norte magnético, y el eje Z es positivo hacia arriba. La orientación calculada por Xsens DOT estará dada por la orientación del sistema S con respecto al L , por lo que los ángulos *roll*, *pitch* y *yaw* serán nulos cuando los ejes x , y , y z del sistema S estén alineados con los ejes X , Y , y Z del sistema L . Para la fase de testeo, el sensor fue colocado en el pie derecho de la persona en prueba, de manera tal que los ejes de coordenada se definen como muestra la Figura 3.22.



Figura 3.21: Sistema de coordenadas local al sensor S



Figura 3.22: Definición del sistema S para Xsens DOT ubicado en el pie derecho

Para almacenar y visualizar los datos obtenidos, se ofrecen varios softwares que se pueden usar en dispositivos móviles o en computadoras. En el presente proyecto, se trabajó con la aplicación móvil *Movella DOT*, donde el usuario simplemente debe conectar el sensor con su dispositivo móvil por Bluetooth para que el sensor empiece a transmitir datos al celular. Luego, se debe elegir la opción de *Real-time Streaming* en la sección de mediciones y recolección de datos, para transmitir y guardar los datos en tiempo real al dispositivo móvil. En el presente proyecto, se eligió trabajar con el modo customizado 1 de los cinco *Custom Modes* que ofrece el sensor, el cual permite medir la aceleración lineal libre (*free acceleration*) en m/s^2 , la orientación en ángulos de Euler en *deg* y la velocidad angular en *deg/s*. Al apretar el botón de *Apply*, el sensor comenzará a medir los tres parámetros y transmitirlos al software en tiempo real, que además se visualizarán en los tres gráficos separados con colores distintos para los ejes de coordenadas tridimensionales. En el Anexo, se indican los pasos para usar la aplicación con imágenes (Figuras 9.9 - 9.13).

3.4. Procesamiento de señales inerciales

En el presente trabajo, se busca calcular la distancia recorrida total, la cantidad de pasos y la cantidad de giros de la persona durante una sesión de entrenamiento con *BrainWalk VR*. El sensor Xsens DOT permitió medir la aceleración libre, la orientación en ángulos de Euler, y la velocidad angular en tres ejes de coordenadas en el sistema L , que luego fueron procesadas para obtener los parámetros deseados. El procesamiento de señales inerciales se realizó en Jupyter Notebook, usando el lenguaje de programación Python.

3.4.1. Cálculo del recorrido total

Para calcular la distancia recorrida total de la persona, se decidió primero aplicar la doble integral discreta a los datos de aceleración lineal y obtener el desplazamiento lineal en los dos ejes X e Y. Los componentes del eje Z, que representan los movimientos verticales del pie, fueron descartados para calcular el recorrido, ya que en este trabajo se busca obtener cuánto recorrió la persona en el plano horizontal (X,Y). Luego de obtener el desplazamiento lineal en dos dimensiones, se calculó la distancia entre cada punto de coordenadas y se realizó la sumatoria de todas las distancias, que representa la distancia recorrida total de la persona.

Previo a cada integración de las señales, se aplicó un filtro de Butterworth de tipo pasaaltos, con orden 4 y su frecuencia de corte en 0,4 Hz. Los parámetros del filtro se basaron en una tesis de grado de Bioingeniería donde se utilizaron los mismos sensores Xsens DOT para obtener el desplazamiento lineal [87]. Este filtro se utilizó para evitar derivas propias de la sumatoria de ruidos de la señal original de aceleración, y se aplicó con la función `signal.butter()` de la librería `scipy` de Python.

Para validar el funcionamiento eficiente del algoritmo para calcular el recorrido, se tomaron seis muestras de una persona caminando una distancia de 3 metros en 6 pasos, en línea recta sobre el plano horizontal. Al aplicar el algoritmo, se obtuvieron las curvas de aceleración, velocidad y desplazamiento lineal en los tres ejes de coordenadas para cada señal de muestra, como se puede ver en la Figura 3.23. Con las seis muestras, la distancia recorrida calculada por el algoritmo fue $2,97 \pm 0,12$ m (valor medio \pm desvío estándar), lo cual es significativamente cercano al recorrido real de 3 metros. De esta manera, se pudo validar el funcionamiento eficiente del algoritmo para calcular la distancia en el plano horizontal.

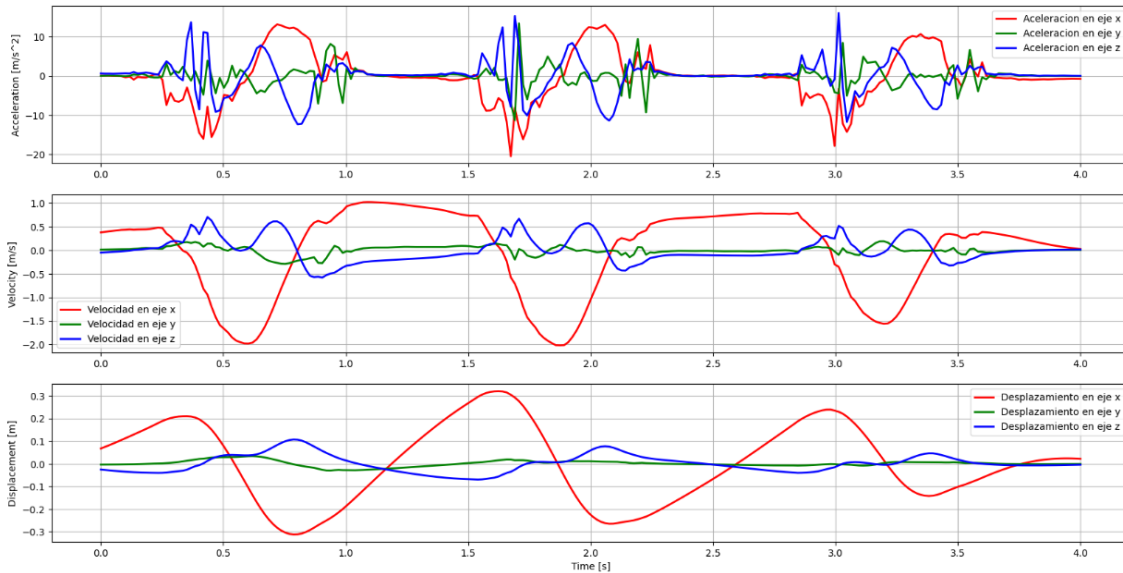


Figura 3.23: Aceleración, velocidad y desplazamiento lineal en los ejes X, Y, Z en una de las seis muestras para la validación del algoritmo

3.4.2. Cálculo de la cantidad de pasos

Para contar la cantidad de pasos realizados por la persona, se decidió trabajar con los datos de velocidad angular en eje Y que representa el movimiento de balanceo en la dirección de la marcha. Un paso en la marcha se define como el acontecimiento entre sucesivos puntos de contacto de pies alternos con el suelo. Si bien en la fase de testeo se colocó el sensor solamente en el pie derecho, se pudieron detectar los pasos del pie izquierdo también, gracias a que la velocidad angular en eje Y presenta dos picos de gran amplitud: uno indica el contacto y el apoyo del pie derecho, mientras que el otro representa la elevación del talón del pie derecho cuando el talón del pie izquierdo toca el piso. El segundo pico aparece ya que el pie derecho se acelera inmediatamente después de que se despeguen sus dedos.

El algoritmo para detectar estos picos de amplitud se diseñó basándose en el método de Pan-Tompkins, que se utiliza para detectar los complejos QRS en las señales de electrocardiograma (ECG). En este método, se aplica un filtro pasabandas para filtrar solamente el rango de frecuencias en que se encuentran los complejos QRS normalmente, y luego se utiliza un filtro derivador para amplificar los momentos del ECG en los que la pendiente es alta, y atenuar el resto. Finalmente, se eleva al cuadrado la señal para tener todos los datos positivos, y se aplica una ventana de integración que tenga duración similar a la de un complejo QRS, para atenuar los ruidos restantes. En este trabajo, bastó con aplicar un filtro pasabandas para atenuar los ruidos y las derivas, y elevar al cuadrado la señal para luego, detectar los picos altos con la función `signal.find_peaks()` de la librería `scipy` de Python.

Para encontrar el rango de frecuencias del filtro pasabandas, se realizó el espectro en frecuencia aplicando la transformada rápida de Fourier a las señales de velocidad angular obtenidas en la fase de testeo (Figura 3.24). En el espectro en frecuencia, se observó que las componentes frecuenciales de la señal se encuentran en su mayoría en el rango entre 0,5 y 4 Hz. Entonces, teniendo en cuenta que el filtro de deriva al calcular el recorrido tenía una frecuencia de corte de 0.4 Hz, y considerando que las señales de interés se encuentran por debajo de 3 Hz, se decidió aplicar un filtro pasabandas entre 0,4 y 3 Hz. Luego de limpiar y suavizar la señal con el filtro, se elevó al cuadrado la señal para obtener todos los valores en positivos y maximizar las amplitudes grandes.

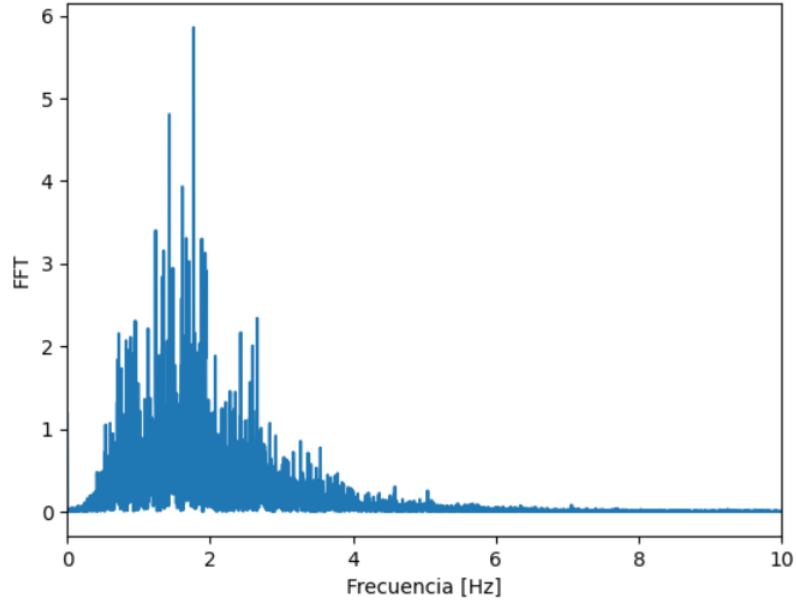


Figura 3.24: Espectro en frecuencia de la señal de velocidad angular en eje Y en una de las 11 muestras obtenidas en el testeo

El algoritmo de detección de pasos se validó con las seis muestras que se obtuvieron caminando 3 metros, que también se utilizaron para validar el algoritmo de cálculo del recorrido total. En la Figura 3.25, se puede observar que la señal se suavizó efectivamente luego de filtrar los ruidos, y al elevar al cuadrado, los picos altos que marcan los pasos de la marcha se amplifican y se marca la diferencia con el resto de la señal.

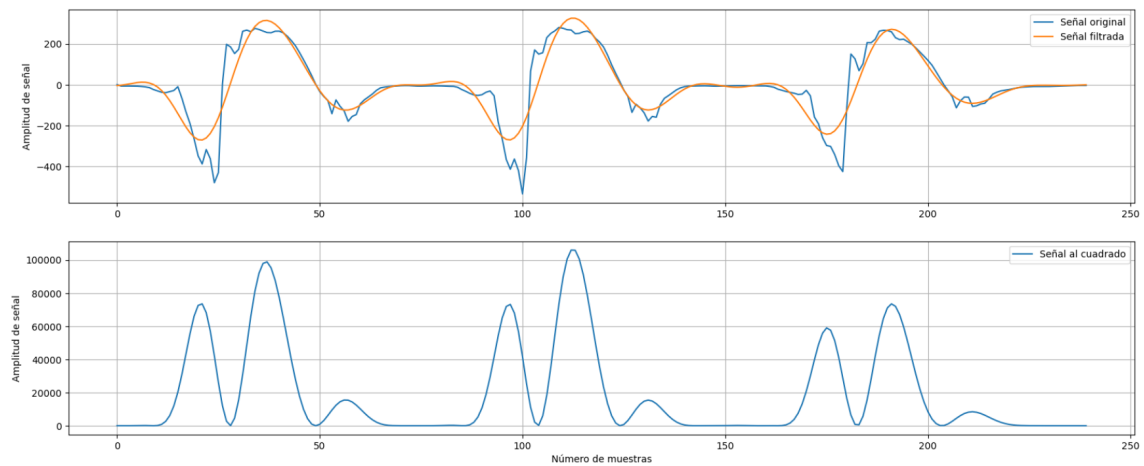


Figura 3.25: Señal original de velocidad angular, señal filtrada y señal elevada al cuadrado en una de las seis muestras para la validación del algoritmo

Luego, el algoritmo usa el umbral de percentil 85 para detectar los picos, es decir que busca

aquellos picos que tienen amplitud mayor al 85 % de los datos de la señal. En la Figura 3.26, se pueden observar los picos que fueron detectados por el algoritmo son efectivamente los picos que representan los pasos de la marcha. Estos picos están agrupados de a par, ya que se producen cuando el pie derecho toca el suelo por primera vez y al levantar su talón cuando el pie izquierdo toca el piso por primera vez. Entonces, en esta señal por ejemplo se detectaron 6 pasos lo cual coincide con la cantidad de pasos que se realizaron en la marcha de 3 metros. Con las seis muestras, el algoritmo detectó 6 pasos con un desvío estándar de 1 paso, lo cual es razonable ya que en una marcha real, la persona no siempre inicia la marcha con el pie derecho, que es donde tiene colocado el sensor inercial. Entonces, por ejemplo si la persona comienza a caminar con el pie izquierdo, este paso no será detectado.

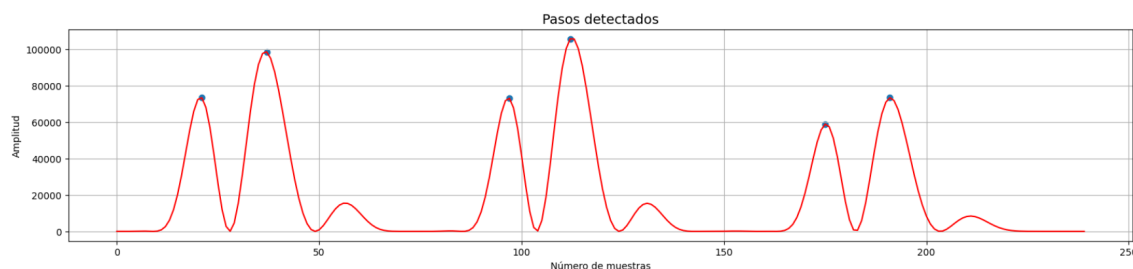


Figura 3.26: Señal procesada con los picos detectados, que representan los pasos de la marcha, en una de las seis muestras para la validación del algoritmo

Es importante destacar el pico pequeño en forma de onda que aparece después del segundo pico alto, ya que representa el movimiento de *toe clearance* donde la persona levanta el pie para no arrastrarlo al alternarse de pie. Por lo tanto, se puede decir que el filtro pasabandas suavizó la señal adecuadamente para limpiar la señal pero conservando las características de la marcha, y también el algoritmo pudo detectar los pares de picos usando el umbral de percentil 85, sin confundirse con los picos de baja amplitud.

3.4.3. Cálculo de la cantidad de giros

En el sensor Xsens DOT, el cambio de orientación del sujeto se define con respecto al sistema local fijo a la Tierra, en el que el eje Z es positivo hacia arriba. Por esta razón, para contar la cantidad de giros realizados por los participantes, se decidió trabajar con los ángulos de Euler en eje Z, que representan los movimientos de rotación que realiza el usuario alrededor del eje vertical al suelo. En las mediciones del sensor, la polaridad de rotación se define por la regla de la mano derecha, entonces apuntando el pulgar a lo largo de la dirección del eje Z, la rotación hacia izquierda será positiva, y hacia derecha será negativa.

Antes de poder contar la cantidad de giros, fue necesario reconstruir las señales porque los datos tenían saltos verticales debido a sus límites entre -180 y 180 grados. Estos saltos de 360 grados eran indeseados porque no permitían determinar cuándo los ángulos seguían aumentando o disminuyendo, por lo que no se lograba obtener la tendencia de la señal. Para detectar estas discontinuidades, se derivó la señal para visualizar los cambios abruptos de gran amplitud que se producen en los saltos. En los gráficos de la Figura 3.27, se puede ver que un salto de -180 a 180 grados en la señal se muestran con un pico positivo en su derivada, mientras que un salto de 180

a -180 grados se manifiesta con un pico negativo en la derivada de la señal. Los puntos donde se detectaron los saltos se marcaron con color rojo en el gráfico, y se guardaron para luego reconstruir la señal, como se ve en el último gráfico en la Figura 3.27.

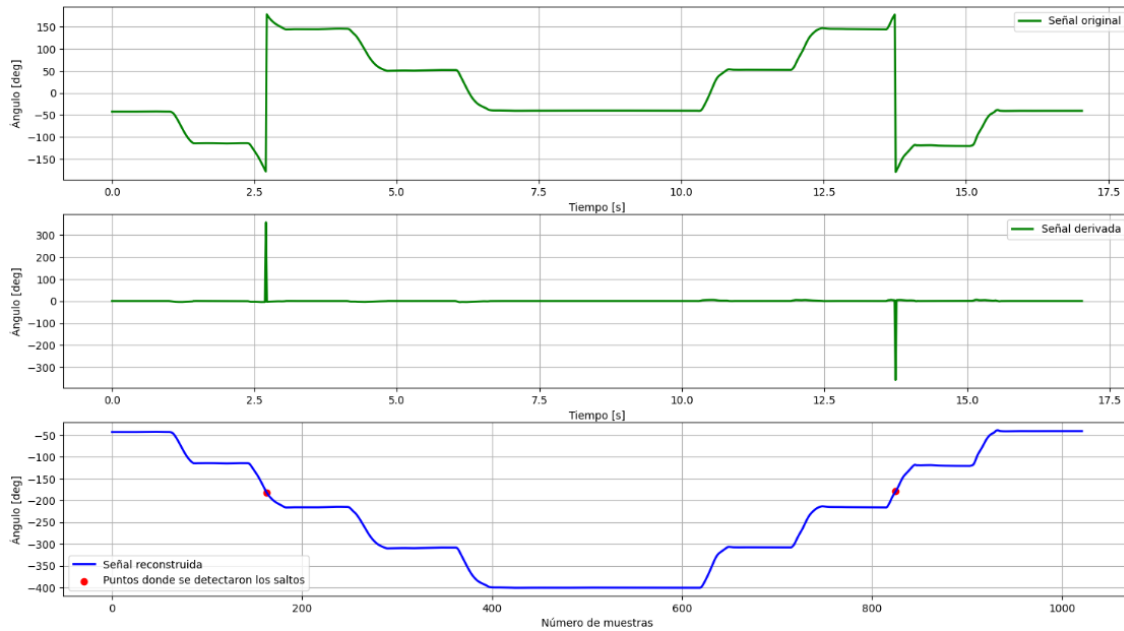


Figura 3.27: Reconstrucción de la señal de ángulos de Euler

Luego de reconstruir las señales, se les aplicó el filtro de Savitzky–Golay que se basa en la regresión polinomial para suavizar la señal preservando características de la distribución inicial tales como los máximos y mínimos relativos, así como el ancho de los picos. Después de obtener las señales suavizadas, se derivaron y se detectaron los picos con la función `signal.find_peaks()` de la librería `scipy` de Python. En el gráfico de la Figura 3.28, se puede ver que los picos negativos en la derivada de señal representan a las disminuciones de ángulos en la señal, mientras que los aumentos de ángulos se muestran con picos positivos en la derivada de señal. Entonces, como la rotación es positiva cuando el sujeto gira hacia la izquierda, los picos positivos representan giros hacia la izquierda y los negativos, giros hacia la derecha.

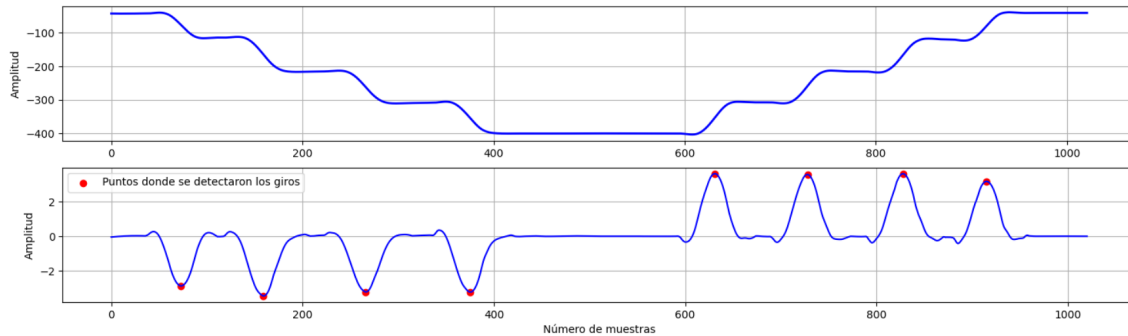


Figura 3.28: Señal reconstruida (arriba) y la detección de giros en la derivada de la señal suavizada de ángulos de Euler (abajo)

Para validar el algoritmo de detección de giros, se tomaron 8 muestras del sujeto girando en diferentes medidas alrededor del eje vertical al suelo:

1. 5 giros de 72 grados hacia la derecha
2. 5 giros de 72 grados hacia la izquierda
3. 4 giros de 90 grados hacia la derecha
4. 4 giros de 90 grados hacia la izquierda
5. 4 giros de 90 grados hacia la derecha, 4 giros de 90 grados hacia la izquierda
6. 8 giros de 45 grados hacia derecha, 8 giros de 45 grados hacia izquierda
7. 2 giros de 180 grados hacia derecha
8. 2 giros de 180 grados hacia izquierda

Los gráficos de las Figuras 3.27 y 3.28 corresponden a la muestra donde la persona realizó 4 giros de 90 grados hacia la derecha, y 4 giros de 90 grados hacia la izquierda. Se puede ver en los gráficos que se detectaron 4 picos negativos y 4 picos positivos, los cuales coinciden con la cantidad de giros y la dirección de cada giro. En las muestras, el algoritmo pudo detectar casi en perfección la cantidad de giros, salvo en la sexta muestra donde no se detectó uno de los giros hacia la derecha debido a su baja intensidad. Entonces, a pesar de que su precisión disminuye cuando el rango de rotación es menor y por lo tanto, la amplitud de señal es menor también, el algoritmo logra detectar la cantidad de giros efectivamente, diferenciando su dirección.

4. Resultados

4.1. Usabilidad del sistema

Para evaluar la usabilidad del sistema, se obtuvieron los puntajes de Sistema de Escala de Usabilidad (SUS) de los participantes luego de realizar una sesión de entrenamiento en *BrainWalkVR*. El valor medio de los puntajes obtenidos fue $80,27 \pm 10,81$ (desvío estándar). Según la tabla de puntajes SUS con sus rangos percentiles (Tabla 5), el valor medio pertenece al rango percentil entre 85 y 96, que corresponde a una calificación «A». Por lo tanto, el software de la aplicación *BrainWalkVR* se evalúa como un sistema con usabilidad «excelente» y «aceptable». Además, obtiene un NPS «Promotor», lo cual indica que los usuarios se encuentran satisfechos con la aplicación, de manera tal que los llevará a convertirse en consumidores fieles y embajadores naturales del producto.

Puntaje SUS	Rango percentil	Calificación	Adjetivo	Aceptabilidad	NPS
[84 - 100]	[96 - 100]	A	Mejor Imaginable	Aceptable	Promotor
[79 - 84)	[85 - 96)	A	Excelente	Aceptable	Promotor
[73 - 79)	[65 - 85)	B	Excelente	Aceptable	Pasivo
[71 - 73)	[60 - 65)	C	Bueno	Aceptable	Pasivo
[63 - 71)	[35 - 60)	C	Bueno	Marginalmente Aceptable	Pasivo
[52 - 63)	[15 - 35)	C	Regular	Marginalmente Aceptable	Detractor

Cuadro 5: Tabla de puntajes SUS con sus respectivos rangos percentiles, calificaciones, escala de adjetivos, aceptabilidad y NPS

El puntaje total de SUS fue mayor a 68 puntos, lo cual indica que se encuentra por arriba del rango percentil 50 del dataset, marcado con una línea horizontal de color rojo en el gráfico de barras en la Figura 4.1. Observando los puntajes individuales de los participantes, se puede ver que de 11 voluntarios, solamente una persona obtuvo el puntaje menor a 68 puntos, lo cual explica el resultado de las pruebas. Además, este puntaje mínimo fue 65 puntos que corresponde a un sistema marginalmente aceptable, con una buena usabilidad, por lo cual se puede decir que los participantes evaluaron positivamente la usabilidad del software.

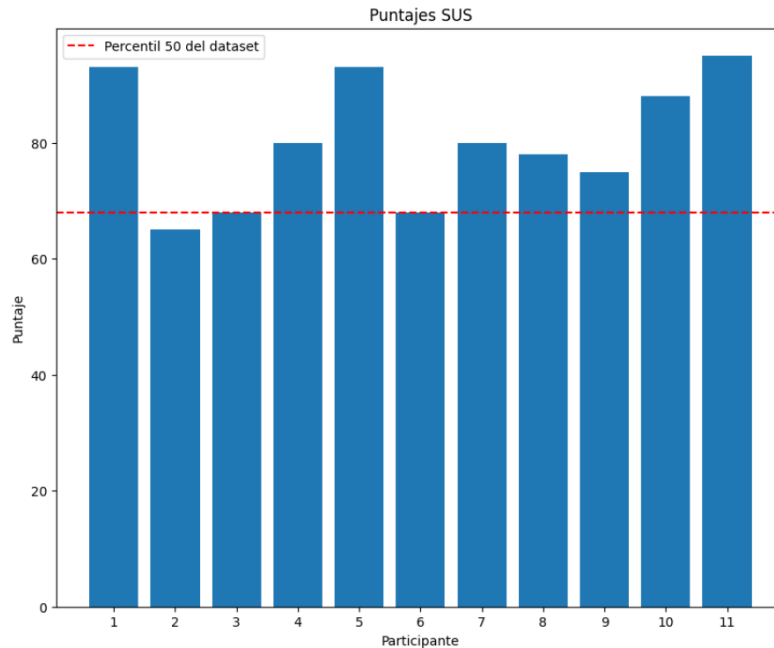


Figura 4.1: Gráfico de barras - Puntajes SUS de los participantes

En cuanto a la distribución de los datos, estuvieron simétricamente distribuidos, ya que el valor medio (80,27) y la mediana (80,0) fueron muy cercanos. Por otro lado, en el gráfico de cajas y bigotes (Figura 4.2) se muestra que la mayoría de los participantes obtuvo puntajes entre 71,5 y 90,5, lo cual corresponde a su rango intercuartil. Teniendo en cuenta que se considera un sistema usable con un puntaje mayor o igual a 68 puntos (marcado con una línea vertical de color rojo en el gráfico de la Figura 4.2), el 75 % de los participantes obtuvieron puntajes mayor al rango percentil 50 del dataset, por lo que se puede decir que el sistema tiene alta usabilidad.

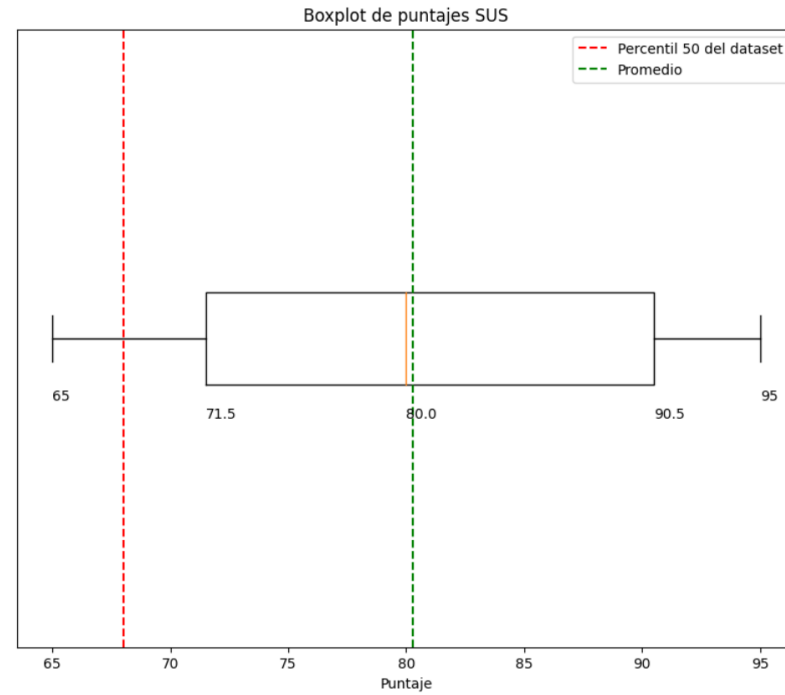


Figura 4.2: Gráfico de cajas y bigotes - Distribución de puntajes SUS de los participantes

4.2. Grado de *cybersickness*

Para evaluar el grado de *cybersickness*, se obtuvieron los puntajes de Cuestionario de Cibermareo en Realidad Virtual de los participantes luego de realizar una sesión de entrenamiento con *BrainWalkVR*, de la misma forma que SUS. El valor medio de los puntajes obtenidos fue $7,27 \pm 1,19$ puntos (valor medio \pm desvío estándar), lo cual corresponde a un grado de *cybersickness* «mínimo» según la escala de evaluación que se armó para la interpretación del puntaje total CSQ-VR. El puntaje mínimo que uno puede obtener en el cuestionario es 6 puntos, lo cual está marcado con una línea punteada de color rojo en el gráfico de la Figura 4.3. Se puede ver que de 11 participantes, 4 personas obtuvieron puntajes mínimos, mientras que el resto de puntajes se encuentran entre 7 y 9 puntos lo cual indica un grado de *cybersickness* mínimo.

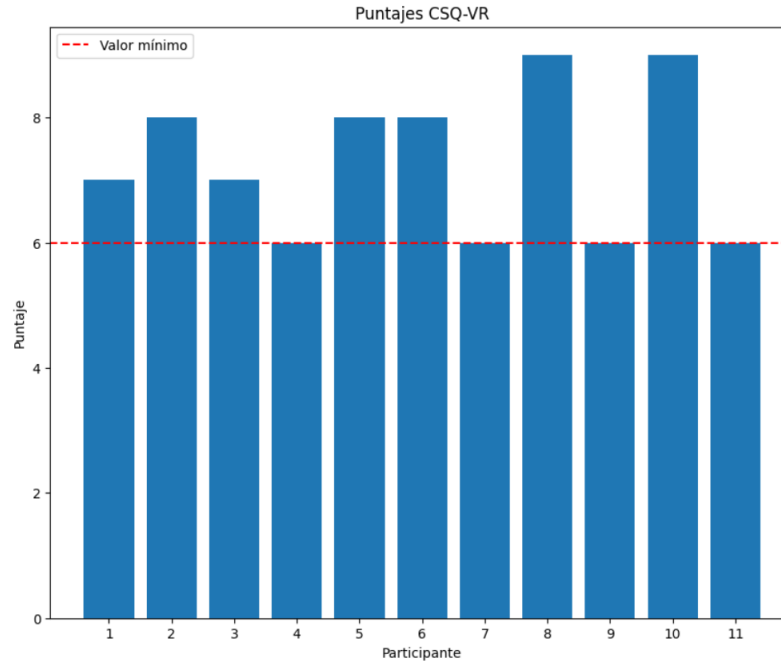


Figura 4.3: Gráfico de barras - Puntajes CSQ-VR de los participantes

Analizando la distribución de los datos, se puede observar que el valor mínimo coincide con el primer cuartil Q1, lo cual significa que el 25 % de los participantes con CSQ-VR más bajo tuvieron el mismo puntaje. Además, teniendo en cuenta que un mayor CSQ-VR indica que el usuario tuvo síntomas de *cybersickness* con mayor intensidad, se puede decir que este 25 % de los participantes tuvo síntomas de *cybersickness* con menor intensidad que el resto. Esto puede ocurrir también porque se trabajaron con número de muestras pequeño, por lo que hay más probabilidad de que haya múltiples valores iguales.

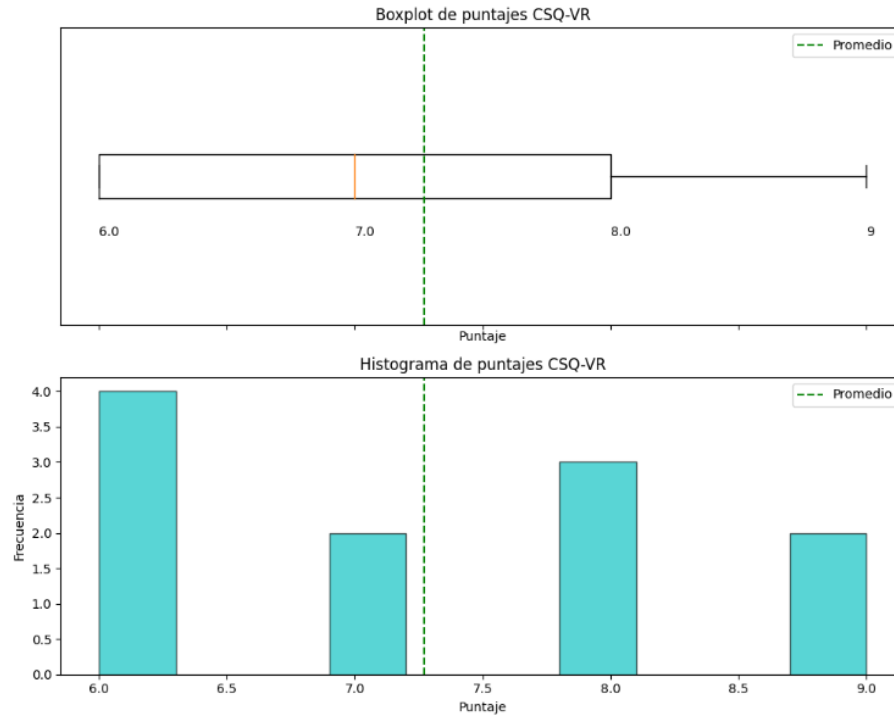


Figura 4.4: Gráfico de cajas y bigotes - Distribución de puntajes CSQ-VR de los participantes

Luego, se analizaron los puntajes por categoría para evaluar el tipo de síntomas de *cybersickness* que más tuvieron los participantes. El puntaje promedio fue: 2,64 puntos para las náuseas, 2,18 puntos para la desorientación, y 2,45 puntos para los síntomas oculomotores. Según la escala de evaluación creada para los puntajes por categoría, estos resultados indican que los participantes tuvieron cada tipo de síntomas en su mínima intensidad. En base a los porcentajes de participantes que obtuvieron un puntaje mayor en cierta categoría, se pudo analizar también el tipo de síntomas de *cybersickness* que tuvieron principalmente durante la prueba. El gráfico de torta en la Figura 4.5 muestra que el 27,3 % de los participantes eligió náuseas como el síntoma principal, mientras que el 18,2 % y 9,1 % eligieron los síntomas oculomotores y vestibulares, respectivamente. Sin embargo, al 45,5 % de los participantes les resultó de igual intensidad los tres tipos de síntomas, por lo que se puede decir que no hubo un tipo de síntomas dominante de *cybersickness* durante los testeos.

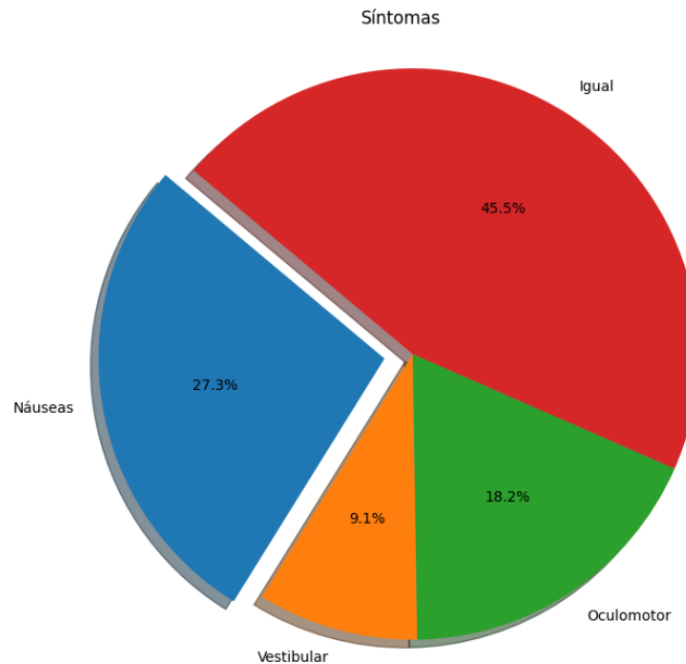


Figura 4.5: Gráfico de torta - Porcentajes de dominancia de un tipo de síntomas de *cybersickness* en los participantes

4.3. Mediciones inerciales

Para calcular el recorrido, la cantidad de pasos y la cantidad de giros realizados durante una sesión de *BrainWalkVR*, se realizaron las mediciones de aceleración lineal, velocidad angular y cambio de orientación en ángulos de Euler con el sensor Xsens DOT, y se obtuvieron 11 muestras.

La distancia recorrida promedio fue $115,18 \pm 21,19m$, con su duración media de 331.36 segundos lo cual equivale aproximadamente a 5 minutos 31 segundos. Considerando los momentos de pausa durante el entrenamiento, este resultado indica que la aplicación permite recorrer una distancia significativa. En la Figura 4.6, se muestran las curvas de aceleración, velocidad y desplazamiento lineal en los tres ejes de coordenadas. Se puede observar que la señal de desplazamiento lineal en eje Z, que representa los movimientos verticales del pie, tiene una amplitud pequeña en comparación con las señales en ejes X e Y, lo cual indica que aporta en menor medida al recorrido total de la persona y justifica la exclusión de los componentes verticales al calcular la distancia recorrida.

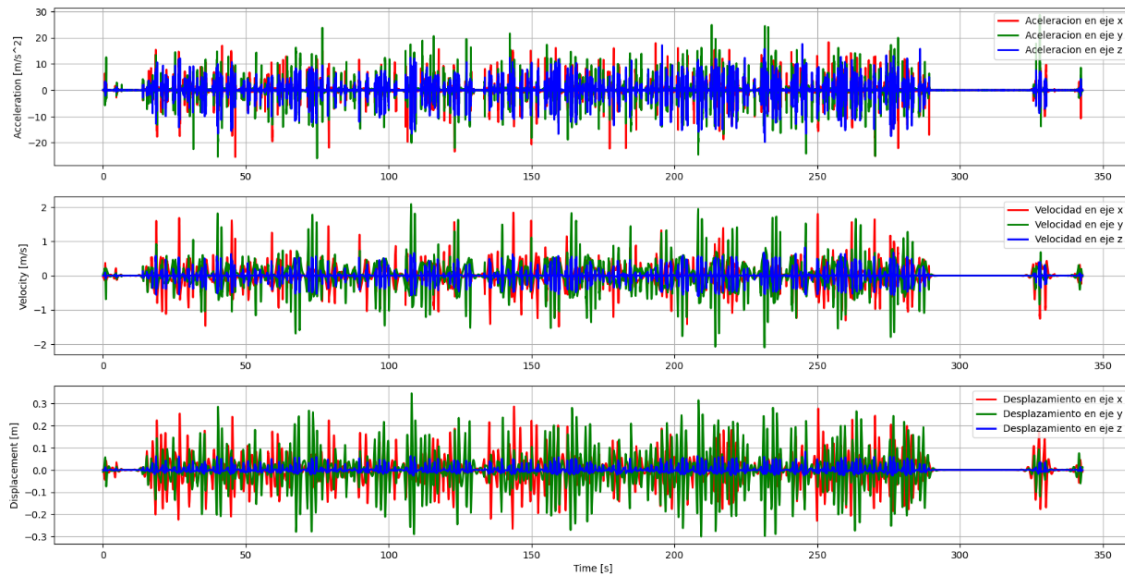


Figura 4.6: Aceleración, velocidad y desplazamiento lineal en los ejes X, Y, Z en una de las 11 muestras obtenidas en el testeo

En el gráfico de cajas de bigotes (Figura 4.7), se puede observar que el primer cuartil Q1 es 105,09 lo cual indica que el 75 % de los participantes recorrieron más o igual que 105,09 metros. Además, en el histograma (Figura 4.7) se puede ver que de 11 participantes solo 2 participantes tuvieron recorrido menor a 100 metros, mostrando que la aplicación exige caminar una distancia significativa durante el entrenamiento.

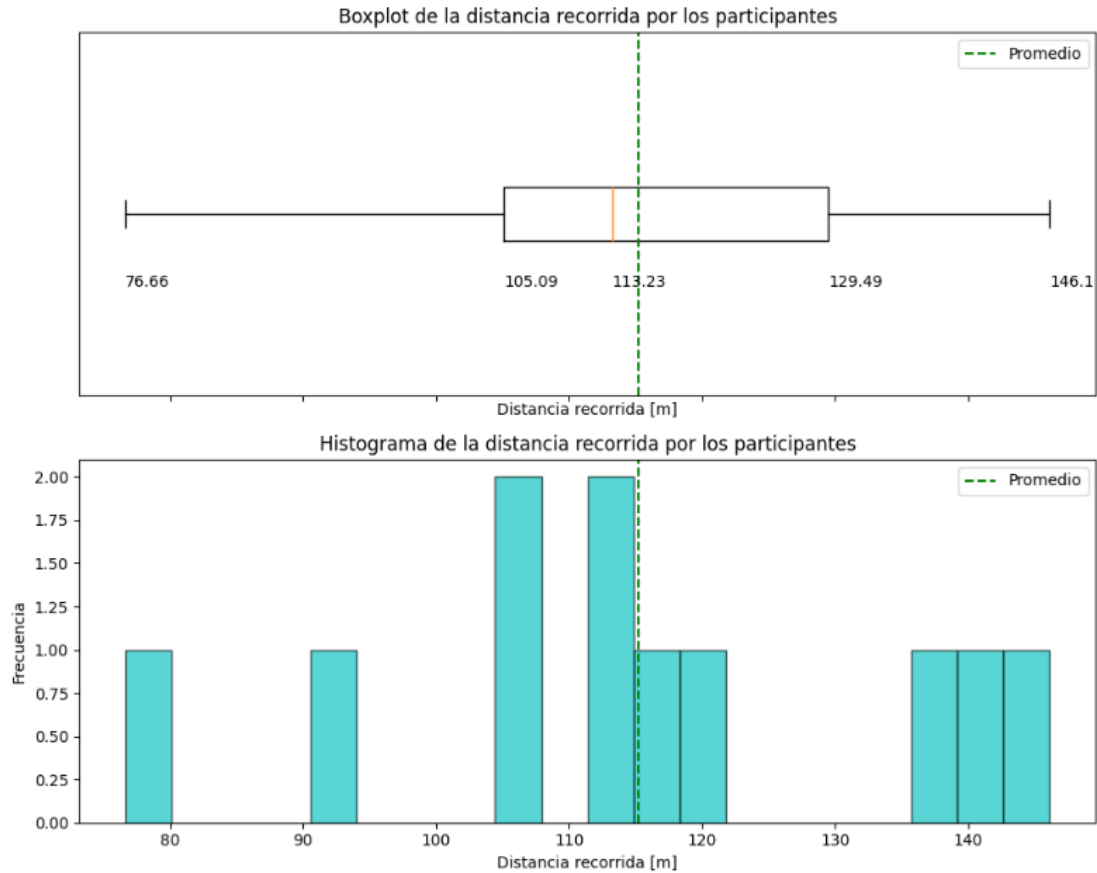


Figura 4.7: Gráfico de cajas y bigotes - distribución de la distancia media recorrida por los participantes

La cantidad de pasos promedio fue 258 ± 57 pasos, con su longitud de paso promedio de $45,69 \pm 7,94\text{cm}$, que se calculó dividiendo la distancia recorrida sobre la cantidad de pasos. Este resultado es razonable, ya que si bien la longitud media del paso para las mujeres adultas es de 67 cm, y la de los hombres, de 76,2 cm [122], los participantes no realizaron una marcha recta durante el testeo, sino que estuvieron dando pasos más pequeños que una caminata normal para recorrer el espacio. En la Figura 4.8 se puede ver la señal de velocidad angular en eje Y en una de las muestras antes y después del procesamiento, y los pasos detectados por el algoritmo, marcados con los puntos en el gráfico de la Figura 4.9.

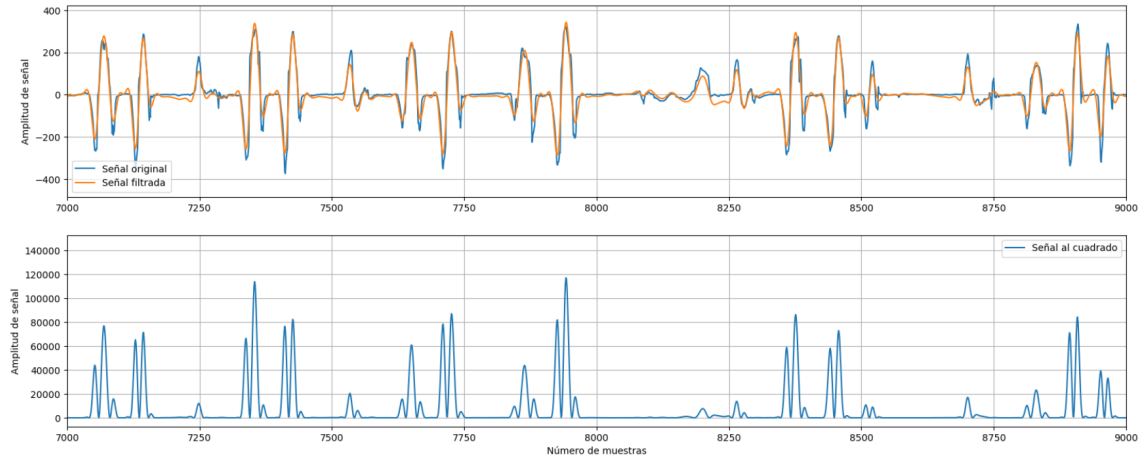


Figura 4.8: Señal original de velocidad angular en eje Y, señal filtrada, y señal elevada al cuadrado en una de las 11 muestras obtenidas en el testeo



Figura 4.9: Señal procesada con los picos detectados en una de las 11 muestras obtenidas en el testeo

En el gráfico de cajas de bigotes (Figura 4.10), se puede observar que el primer cuartil Q1 es 215 lo cual indica que la cantidad de pasos del 75 % de los participantes fue mayor o igual que 215 pasos. Además, en el histograma (Figura 4.10) se puede ver que 10 de 11 participantes caminaron más de 200 pasos durante el entrenamiento, lo cual muestra que la aplicación exige un rendimiento físico significativo durante la realización de actividades en *BrainWalkVR*.

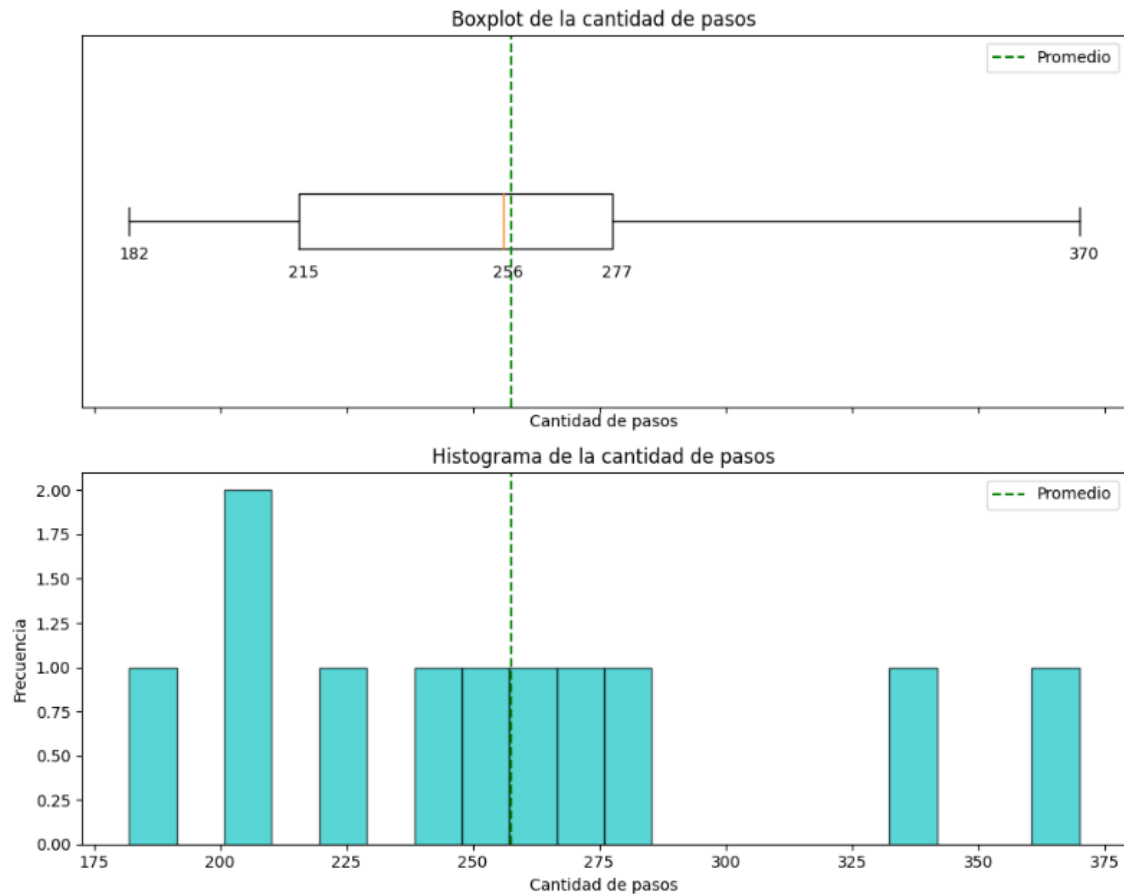


Figura 4.10: Gráfico de cajas y bigotes - distribución de la cantidad de pasos realizados por los participantes

La cantidad de giros promedio que realizaron los participantes durante el entrenamiento fue 166 ± 46 giros, con 73 giros hacia derecha y 93 giros hacia izquierda. En el gráfico de la Figura 4.11, se puede ver una de las señales de ángulos de Euler que se obtuvieron en los testeos y los picos detectados en la derivada de señal, que representan los giros hacia la derecha si son negativos, y hacia la izquierda si son positivos. Se observa también que existen tantos picos positivos como negativos, lo que indica que existe un equilibrio mediano entre los giros hacia la derecha y hacia la izquierda, tal como muestran los resultados numéricos.

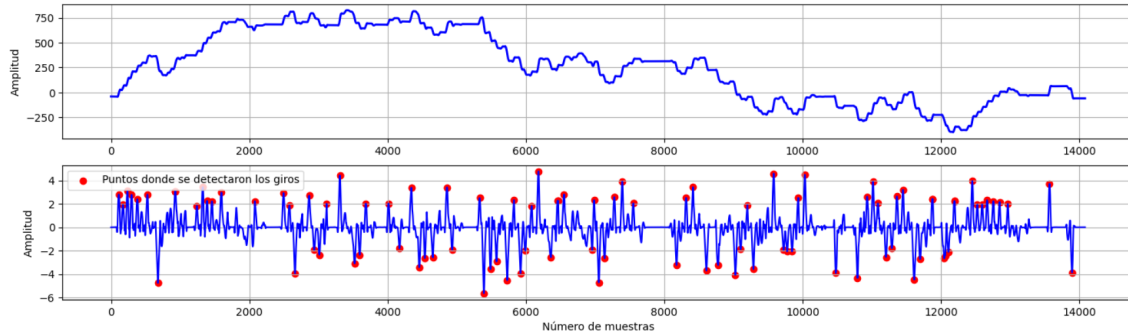


Figura 4.11: Señal de ángulos de Euler con los picos detectados en una de las 11 muestras obtenidas en el testeo

En el gráfico de cajas de bigotes (Figura 4.12), se puede observar que el primer cuartil Q1 es 147 lo cual indica que la cantidad de pasos del 75 % de los participantes fue mayor o igual que 147 pasos. Además, en el histograma (Figura 4.12) se puede ver que 9 de 11 participantes realizaron más de 140 giros durante el entrenamiento, lo cual muestra que la aplicación logra exigir al usuario realizar una cantidad de giros significativa durante la realización de actividades en *BrainWalkVR*.

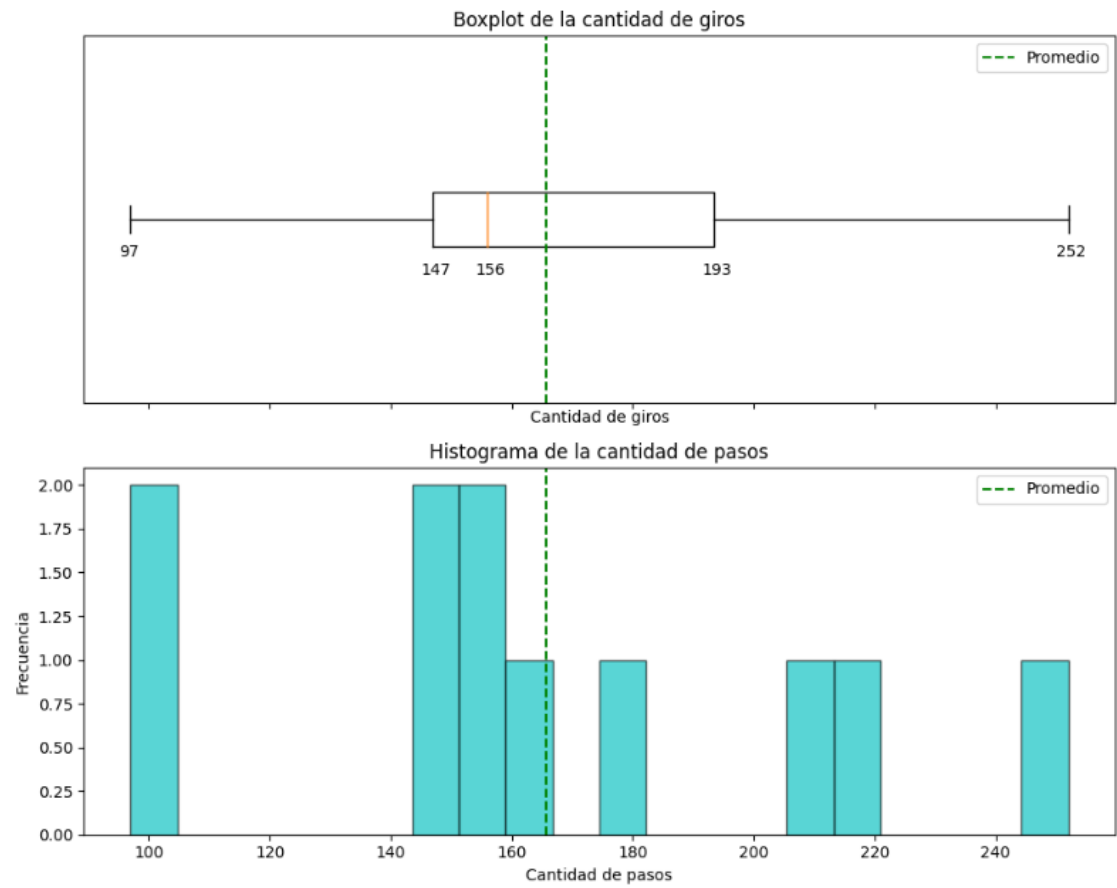


Figura 4.12: Gráfico de cajas y bigotes - distribución de la cantidad de giros realizados por los participantes

5. Discusión

En la fase de testeo, se obtuvieron resultados relacionados con el desarrollo de la aplicación de entrenamiento en realidad virtual, *BrainWalkVR*, para pacientes con Parkinson. Se evaluaron la usabilidad del sistema, el grado de *cybersickness* que produce la aplicación, y el rendimiento físico que se haya logrado durante una sesión de entrenamiento, para una población sana. Estos resultados proporcionan una base sólida para comprender el potencial de esta aplicación en la rehabilitación de pacientes con Parkinson.

5.1. Usabilidad del sistema desde la perspectiva de los usuarios

Comenzando con la evaluación de usabilidad, es fundamental analizar cómo los participantes interactuaron con la aplicación y qué desafíos se presentaron. El valor medio de los puntajes obtenidos en el Sistema de Escala de Usabilidad (SUS) fue $80,27 \pm 10,81$ puntos con el desvío estándar, lo cual le otorga una calificación «A» que indica un sistema con usabilidad «excelente» y «aceptable», según la tabla 5. Según el gráfico de barras en la Figura 4.1, 10 de 11 participantes obtuvieron un puntaje mayor o igual a 68 puntos, lo cual indica que la usabilidad del sistema es mayor a la usabilidad promedio del dataset en el cual se basa la comparación estadística. Además, en el gráfico de cajas y bigotes en la Figura 4.2, se puede ver que el 75 % de los participantes obtuvieron puntajes mayores a 71,5 puntos, lo cual corresponde al primer cuartil Q1. Es importante reconocer las limitaciones de este resultado estadístico ya que se trabajó con un número pequeño de muestras en la fase de testeo. A pesar de las limitaciones, este resultado muestra que la mayoría de los participantes pudieron interactuar con la aplicación de manera eficiente y efectiva, y con un alto grado de satisfacción al usar el sistema.

En el estudio de investigación, *Design and Development of a Virtual Reality-Based Mobility Training Game for People With Parkinson's Disease* [7], se diseñó también una aplicación en realidad virtual, de la cual surgió la idea de diseñar la aplicación *BrainWalkVR*. En el estudio, nueve pacientes testearon el programa a lo largo de tres días, con una sesión de entrenamiento por día. Entonces, evaluaron la usabilidad del sistema a lo largo de tres días con el Sistema de Escala de Usabilidad (SUS), que resultó ser 82,5, lo cual se asemeja a la usabilidad de la aplicación *BrainWalkVR*. Entonces, se puede decir que la aplicación alcanzó un nivel de usabilidad comparable al observado en el estudio en que se fundamentó el proyecto, lo cual sugiere un diseño efectivo y una experiencia de usuario positiva para los usuarios finales.

En el diseño de la aplicación, una de las características que más se consideraron fue la facilidad de aprendizaje, lo cual implica que los usuarios deben poder comprender rápidamente cómo funciona el sistema sin requerir una curva de aprendizaje significativa, para luego poder interactuar con el sistema una vez que hayan aprendido a usarlo. Este elemento es fundamental, teniendo en cuenta que la mayoría de la población con Parkinson pertenece a un rango de edad mayor a 50 años, lo que puede limitarlos a aprender y manejar en forma fluida una tecnología nueva como la realidad virtual. En la aplicación, el usuario solo debe acordarse de la funcionalidad de tres botones para realizar las tareas en todo el sistema, lo cual facilita y acelera el proceso de aprendizaje. Esto se reflejó durante la fase de testeo, donde los participantes no necesitaron más de 5 minutos para escuchar la explicación y aprender cómo usar el dispositivo de RV. Además, es importante destacar

que ninguno de los participantes tenía experiencia previa con RV, lo cual respalda el hecho de que la aplicación fue diseñada con facilidad de aprendizaje y de uso.

Además, la calificación de los puntajes obtenidos en el Sistema de Escala de Usabilidad (SUS) corresponde a un Net Promoter Score (NPS) «promotor», lo cual indica que los usuarios se encuentran satisfechos con la aplicación, en tal medida que los llevará a convertirse en consumidores fieles y embajadores naturales de la aplicación. Si bien NPS es una métrica usada mayormente para medir el nivel de satisfacción del cliente con los productos o servicios de una empresa, es un indicador confiable para evaluar el grado de satisfacción de los participantes al usar la aplicación en los testeos. El resultado muestra que los usuarios realizaron la sesión de entrenamiento con un alto nivel de satisfacción, y esto se puede relacionar con que el sistema es visualmente atractivo y se asemeja a las interacciones en el mundo real. Por ejemplo, la interfaz de usuario como los menús se diseñó en forma tal que sea intuitivo usarla, y en el lugar de los controladores el usuario puede ver las manos virtuales que realicen animaciones de acuerdo a las acciones del usuario; por ejemplo, cuando el usuario agarra una pieza, el objeto y la mano se posicionan en una forma que corresponde a cuando uno sujeta un objeto en vida real.

Por otro lado, los usuarios tenían posibilidad de elegir uno de los tres escenarios del bosque que proporciona la aplicación: soleado, lluvioso y nevado. De entre los 11 participantes, 7 optaron por entrenar en un entorno soleado, mientras que los otros 4 seleccionaron las opciones de escenario lluvioso o nevado, con 2 personas cada uno. La usabilidad del sistema que evaluó cada participante no fue influenciada por el escenario que eligió. Al ofrecer distintas opciones de escenarios, el programa logra adaptarse más a la preferencia personal de cada usuario y por ende, aumentar su motivación y satisfacción al realizar el entrenamiento. Además, al simular un espacio abierto con elementos naturales, incluyendo la presencia de árboles, lluvia y nieve, y al ambientar con diferente música de fondo según el escenario, el sistema permite al usuario sentirse más inmerso en el entorno virtual. Este resultado es esencial para impulsar la motivación de los pacientes a comprometerse con el entrenamiento utilizando *BrainWalkVR*.

5.2. Grado de *cybersickness* experimentado por los usuarios

La percepción de *cybersickness* por parte de los usuarios es otro aspecto crítico a evaluar para la efectividad de la aplicación. El valor medio de los puntajes obtenidos en el Cuestionario de Cibermareo en Realidad Virtual (CSQ-VR) fue $7,27 \pm 1,19$ puntos con el desvío estándar, lo cual corresponde a un grado de *cybersickness* «mínimo», según la escala de evaluación (Tabla 3). En el gráfico de barras en la Figura 4.3, se puede ver que 4 de 11 participantes obtuvieron el puntaje mínimo de 6 puntos, lo cual indica que aproximadamente el 25 % de usuarios no experimentaron ninguno de los síntomas relacionados al *cybersickness* durante el entrenamiento con la aplicación *BrainWalkVR*. Este resultado se visualiza también en el gráfico de cajas y bigotes en la Figura 4.4, donde el primer cuartil Q1 coincide con el valor mínimo del puntaje CSQ-VR. Sin embargo, trabajar con un número de muestras pequeño puede aumentar la probabilidad de que haya múltiples valores iguales, lo cual puede influir en el resultado estadístico.

Durante el uso del dispositivo de RV en *BrainWalkVR*, los usuarios experimentaron los síntomas de *cybersickness* en su grado mínimo. Esto se pudo lograr gracias a poca latencia del HP Reverb G2 que se utilizó para la fase de testeo del proyecto. El *cybersickness* depende fuertemente de la

latencia, que es el tiempo de demora entre lo que el sistema de RV entrega y lo que el cerebro espera percibir. Este tiempo de retraso se define por la frecuencia de actualización del dispositivo, que en el caso de HP Reverb G2 es 90 Hz, lo cual indica que actualizará la pantalla 90 veces por segundo. Entonces, al aumentar la frecuencia de actualización, también mejora la suavidad y la velocidad de cambio del movimiento en la pantalla, por lo que habrá menor latencia y por ende, el usuario sufrirá de *cybersickness* con menor intensidad. Además, el dispositivo usado tiene una alta resolución visual de 2160×2160 píxeles por ojo, lo cual ayuda a disminuir la fatiga ocular.

La mayoría de los síntomas de *cybersickness* tiene duración corta y bajo riesgo, cuando la exposición del usuario a la RV es durante cortos períodos de tiempo, pero cuando la exposición se prolonga, su intensidad y duración pueden incrementar. El tiempo medio que tardaron los participantes en finalizar una sesión de entrenamiento fue 331,36 segundos, lo cual equivale aproximadamente a 5 minutos 31 segundos, por lo que su exposición no fue prolongada. Esto podría haber favorecido a baja presencia de síntomas de *cybersickness* en esta fase de testeo. Entonces, a futuro será crítico evaluar los síntomas de *cybersickness* luego de realizar el entrenamiento con *BrainWalkVR* por períodos de tiempo más largos.

El Cuestionario de Cíbermareo en Realidad Virtual (CSQ-VR) analiza los síntomas de *cybersickness* experimentados por el usuario en tres bloques: náuseas, efectos relacionados a la desorientación, y las molestias y fatiga oculomotora. Los puntajes promedio obtenidos en la fase de testeo para cada categoría fueron: 2,64 puntos para las náuseas, 2,18 puntos para la desorientación, y 2,45 puntos para los síntomas oculomotores. Según la escala de evaluación (Tabla 4), estos resultados indican que los participantes tuvieron cada tipo de síntomas en su mínima intensidad. En términos de porcentajes, al 45,5 % de los participantes les resultó indistinta la intensidad de los síntomas de cada categoría. Sin embargo, el 27,3 % eligió las náuseas como los síntomas de mayor intensidad que experimentaron durante el entrenamiento, lo cual es razonable ya que las náuseas son los síntomas más comunes y generales de *cybersickness*, en comparación con los problemas vestibulares y oculomotores, que son de mayor gravedad pero menos frecuentes.

Por otro lado, el 18,2 % eligió los efectos oculomotores como de mayor intensidad: esto puede deberse a que de 11 participantes, 4 personas usaron sus anteojos personales durante las pruebas, entonces más del 25 % de los participantes tuvieron que colocarse el casco arriba de los anteojos. Por lo tanto, estos usuarios podrían haber experimentado una mayor fatiga ocular durante la realización del testeo. Además, existe la posibilidad de que el casco no estuviera ajustado lo suficiente para proporcionar la comodidad necesaria y también para lograr un enfoque adecuado de los visores del casco. Teniendo en cuenta que la mayoría de los pacientes con Parkinson pertenecen a un rango de edad mayor, hay mayor probabilidad de que usen anteojos por lo que será importante considerar el uso de anteojos a la hora de ofrecer un tratamiento de rehabilitación con realidad virtual.

En el estudio de investigación en que se basó la aplicación, *Design and Development of a Virtual Reality-Based Mobility Training Game for People With Parkinson's Disease* [7], nueve pacientes realizaron la actividad a lo largo de tres días, con una sesión de entrenamiento por día. Entonces, se evaluaron los síntomas asociados con el *cybersickness* a lo largo de estos tres días, con el Cuestionario de malestar asociado al simulador (SSQ). En este estudio, no se observaron efectos adversos en medida significativa, aunque dos pacientes presentaron más síntomas de náuseas al segundo día. Este resultado se puede comparar con que en la fase de testeo del presente trabajo, el 27,3 % eligió

las náuseas como los síntomas de mayor intensidad que experimentaron durante el entrenamiento. A futuro, para evaluar el impacto del entrenamiento con la aplicación *BrainWalkVR*, se podría realizar un seguimiento a largo plazo como se hizo en el estudio de investigación.

5.3. Rendimiento físico de los usuarios

El análisis de las señales inerciales proporcionó información valiosa sobre el rendimiento físico de los participantes mientras utilizaron la aplicación de entrenamiento *BrainWalkVR*. Mediante el procesamiento de señales inerciales, se obtuvieron el recorrido total, la cantidad de pasos y la cantidad de giros realizados por los participantes durante el entrenamiento. Estos datos se utilizaron para evaluar el potencial de la aplicación en la rehabilitación físico-funcional de los pacientes con Parkinson. No fue posible comparar los resultados con el estudio de investigación en el que se fundamentó el proyecto [7], ya que no fueron sus criterios de evaluación. Por lo tanto, resulta esencial cuantificar el rendimiento físico que la aplicación logra exigir al usuario, con el fin de obtener una comprensión más precisa de su impacto y eficacia en el entrenamiento físico de los pacientes con Parkinson.

La distancia media recorrida por 11 participantes durante el entrenamiento fue $115,18 \pm 21,19$ metros, con duración media de 331,36 segundos. Considerando los momentos de pausa y que no es un entrenamiento de pura marcha, este resultado indica que la aplicación exige recorrer una distancia significativa en un tiempo razonable. Además, el espacio del entrenamiento tiene una dimensión de 3×3 metros cuadrados, entonces dadas las limitaciones espaciales, se puede decir que la aplicación es capaz de generar un rango de movimiento efectivo dentro de un espacio físico relativamente pequeño. Por otro lado, el histograma en la Figura 4.7 muestra que 4 de 11 participantes caminaron más que 120 metros durante el entrenamiento, entonces teniendo en cuenta que una cuadra en la ciudad de Buenos Aires mide aproximadamente 129,9 metros [123], significa que una sesión de entrenamiento logra que el usuario camine una distancia comparable a la de una cuadra. Este resultado sugiere que la aplicación proporcionó a los participantes una experiencia de entrenamiento lo suficientemente desafiante en términos de distancia recorrida, lo que es un logro importante en el contexto del entrenamiento para pacientes con Parkinson. Además, el resultado podría ser particularmente relevante en entornos clínicos o domiciliarios donde el espacio disponible para el ejercicio es limitado.

La cantidad de pasos promedio realizados por los participantes durante el entrenamiento fue 258 ± 57 pasos, con una longitud promedio de $45,69 \pm 7,94$ centímetros. Esta longitud de paso es razonable ya que si bien la longitud media del paso para las mujeres adultas es de 67 cm y la de los hombres es de 76,2 cm, los participantes no realizan una marcha recta durante el entrenamiento, sino que estuvieron recorriendo con pasos más pequeños que una marcha normal en línea recta. La limitación del resultado es el hecho de que se haya utilizado el sensor inercial solo en el pie derecho, entonces es posible que no se hayan detectado los pasos en caso de que el usuario camina primero con el pie izquierdo. Sin embargo, esta limitación provocaría una subestimación de la cantidad de pasos realizados por los participantes, lo cual no sobrevaloraría el nivel del rendimiento físico que genera esta aplicación de entrenamiento. Por otro lado, teniendo en cuenta la medida de una cuadra en la ciudad de Buenos Aires mencionada anteriormente, una mujer adulta y un hombre adulto darían 194 pasos y 170 pasos respectivamente, para caminar una cuadra aproximadamente. Entonces, el resultado de la cantidad de pasos es razonable considerando que los participantes

dieron pasos más pequeños en un espacio físico limitado durante el entrenamiento. Además, en el histograma en la Figura 4.10 se puede ver que 10 de 11 participantes dieron más o igual que 200 pasos en total, lo que muestra la capacidad de la aplicación para proporcionar a los usuarios un nivel de entrenamiento significativo a pesar de las restricciones del espacio físico.

La cantidad de giros promedio realizados por los participantes durante el entrenamiento fue 166 ± 46 giros, con 73 giros hacia derecha y 93 giros hacia izquierda. La rotación es uno de los movimientos donde los pacientes con Parkinson presentan dificultades, debido a la necesidad de coordinar los movimientos de miembros inferiores y a la rigidez muscular. Por esta razón, es esencial realizar ejercicios de rotación diariamente, lo cual no permiten los tratamientos de marcha sobre la cinta de correr. Los resultados de los testeos muestran que la aplicación BrainWalkVR tiene la capacidad para exigirle al usuario realizar una cantidad de giros significativa, entonces podría ser una alternativa para realizar ejercicios tanto de marcha como de rotación de manera efectiva. Además, en el histograma (Figura 4.12) se puede ver que 9 de 11 participantes realizaron más de 140 giros durante el entrenamiento, lo que muestra su potencial para servir como herramienta para rehabilitación de pacientes con Parkinson.

6. Conclusiones

En el presente proyecto final de grado, se logró desarrollar una aplicación de entrenamiento en realidad virtual, llamada *BrainWalkVR*, para la rehabilitación físico-funcional de pacientes con enfermedad de Parkinson. Además de ser un tratamiento innovador y dinámico, distinto a las terapias convencionales, este trabajo implica un avance en área de rehabilitación en nuestro país, al brindar una solución asequible frente a opciones de alto costo producidas en el exterior.

Para realizar este trabajo, fue necesario utilizar conocimientos adquiridos durante la carrera de Bioingeniería, y también desarrollar otros como el diseño de realidad virtual. El proyecto implicó comprender síntomas complejos de la enfermedad de Parkinson, utilizando los conceptos de anatomía y fisiología. El trabajo abarcó también fundamentos de matemática, física, estadística y procesamiento de señales a la hora de analizar los resultados obtenidos en la fase de testeo.

El mayor desafío se presentó al combinar los tratamientos de rehabilitación convencionales con la tecnología de realidad virtual, para diseñar un entrenamiento más dinámico y motivador para los pacientes, sin comprometer la efectividad de rehabilitación. Considerando los resultados obtenidos en los testeos, se concluye el cumplimiento de los objetivos del proyecto.

En primer lugar, se logró desarrollar un simulador de realidad virtual como herramienta de rehabilitación para pacientes con Parkinson. Para evaluar el rendimiento físico generado por la aplicación, se realizó un análisis de señales inerciales de los participantes durante las sesiones del entrenamiento, donde se pudo calcular la distancia recorrida total media, la cantidad de pasos media y la cantidad de giros media entre las 11 muestras. Los resultados muestran que una sesión de entrenamiento permite caminar en medida significativa, superando las limitaciones que da el espacio físico. Además, a diferencia de los tratamientos en cinta de correr, la aplicación *BrainWalkVR* permite también realizar movimientos de rotación, lo cual es fundamental para tratar la rigidez postural y muscular que sufren los pacientes con Parkinson.

El desarrollo de la aplicación fue acompañado con un proceso de validación interna continua, que consistió en asegurar el cumplimiento de los objetivos terapéuticos en base a las investigaciones con bibliografía e interactuando con los especialistas interdisciplinarios como neurocirujano e ingenieros biomédicos. Luego, para la validación externa, se seleccionó un grupo de población sana cuya edad esté en el rango de aparición de primeros síntomas de enfermedad de Parkinson. A través del Sistema de Escala de Usabilidad (SUS), se evaluó la usabilidad del sistema desde la perspectiva de los usuarios y resultó que la aplicación es sumamente usable, indicando que es fácil de aprender y usar, y que genera un alto nivel de satisfacción en los usuarios.

En cuanto al *cybersickness*, se evaluó el grado de síntomas relacionados que los participantes presentaban después de las pruebas. Los resultados del Cuestionario de Cibermareo en Realidad Virtual (CSQ-VR) mostraron que la intensidad de estos síntomas era prácticamente imperceptible. Esto se logró gracias a alta frecuencia de actualización del modelo del dispositivo de RV que se utilizó para los testeos, que genera menor latencia en la producción de imágenes en el entorno virtual.

Además de abordar las necesidades terapéuticas, el diseño de la aplicación se centró en ofrecer una experiencia de usuario inmersiva y personalizada. Para este desafío, se desarrollaron diferentes escenarios usando varios modelos 3D de elementos de naturaleza, para que el usuario tenga

opciones para elegir según su preferencia. Esta personalización logra aumentar las motivaciones del usuario para volver a usar la aplicación, y también incrementar el nivel de satisfacción. En área de rehabilitación, el factor de motivación es esencial para que los pacientes se comprometan con los tratamientos a largo plazo, por lo que la aplicación se vuelve una potencial herramienta rehabilitadora para pacientes con Parkinson.

Comparando con el estudio de investigación en el que se basó el proyecto, *Design and Development of a Virtual Reality-Based Mobility Training Game for People With Parkinson's Disease* [7], se pudo ver que la aplicación *BrainWalkVR* alcanzó un nivel de usabilidad comparable al observado en el estudio. Además, hubo poca presencia de síntomas asociados con el *cybersickness*, similar a lo observado en el estudio. Ya que el estudio no usó los mismos criterios de evaluación, no se pudo comparar los resultados sobre el rendimiento físico; sin embargo, sirvió para obtener una comprensión más precisa de su impacto y eficacia en el entrenamiento de los pacientes.

Este proyecto también plantea tanto limitaciones como oportunidades futuras. En la fase de testeo, se tomaron un número pequeño de muestras, lo que podría haber afectado la robustez de los resultados estadísticos. Para más adelante, será beneficioso realizar pruebas con un grupo más amplio de participantes para obtener conclusiones más sólidas. Además, se podría considerar la implementación de un seguimiento a largo plazo para evaluar el impacto a largo plazo del entrenamiento con la aplicación *BrainWalkVR*. En el futuro, será fundamental explorar cómo la aplicación puede integrarse de manera efectiva en la práctica clínica, involucrando a médicos y terapeutas especializados en el tratamiento del Parkinson.

En conclusión, el proyecto de desarrollo de la aplicación de entrenamiento en realidad virtual *BrainWalkVR* representa un paso significativo en la dirección de la rehabilitación física y funcional de pacientes con enfermedad de Parkinson. Además de ser una alternativa terapéutica novedosa y motivadora, este proyecto contribuye a la accesibilidad a tratamientos efectivos y asequibles a nivel nacional. Las mejoras continuas y la colaboración con profesionales médicos y terapeutas especializados se presenta como un paso prometedor en la integración de la aplicación en la práctica clínica.

7. Bibliografía

- [1] Dorsey E Ray et al. «The Emerging Evidence of the Parkinson Pandemic». En: *J Parkinsons Dis* 8.s1 (2018), S3-S8. DOI: <https://doi.org/10.3233/jpd-181474>.
- [2] *En Argentina cien mil personas viven con Parkinson*. Fecha de consulta: 03.04.2023. telam. URL: <https://www.telam.com.ar/notas/202204/589136-parkinson-dia-mundial-enfermos-argentina.html>.
- [3] Cremaschi F et al. *Enfermedad de Parkinson: Últimas Etapas*. Editorial Tektime, 2020.
- [4] *¿Qué es la estimulación cerebral profunda y por qué es un tratamiento eficaz para el Parkinson?* Fecha de consulta: 14.08.2023. infobae. URL: <https://www.infobae.com/salud/ciencia/2020/11/05/que-es-la-estimulacion-cerebral-profunda-y-por-que-es-un-tratamiento-eficaz-para-el-parkinson/>.
- [5] Lozano AM et al. «Deep brain stimulation: current challenges and future directions». En: *Nat Rev Neurol*. 15.3 (2019), págs. 148-160. DOI: <https://doi.org/10.1038/s41582-018-0128-2>.
- [6] *Headset technology is cheaper and better than ever*. Fecha de consulta: 03.04.2023. The Economist. URL: <https://www.economist.com/technology-quarterly/2020/10/01/headset-technology-is-cheaper-and-better-than-ever>.
- [7] Finley J. M. et al. «Design and Development of a Virtual Reality-Based Mobility Training Game for People With Parkinson's Disease». En: *Frontiers in neurology* 11 (2021). DOI: <https://doi.org/10.3389/fneur.2020.577713>.
- [8] *Apuntes de la cátedra 16.03 Fisiología*. Fecha de consulta: 30.07.2023.
- [9] *¿Cuáles son las partes del sistema nervioso?* Fecha de consulta: 18.07.2023. Los Institutos Nacionales de Salud. URL: <https://espanol.nichd.nih.gov/salud/temas/neuro/informacion/partes>.
- [10] Fecha de consulta: 18.07.2023. URL: <https://homomedicus.com/cuales-son-las-partes-del-encefalo/>.
- [11] *La sustancia blanca y la sustancia gris del cerebro: función y comparativa*. Fecha de consulta: 18.07.2023. PsicoActiva. URL: <https://www.psicoactiva.com/blog/la-sustancia-blanca-la-sustancia-gris-del-cerebro-funcion-comparativa/>.
- [12] Fecha de consulta: 18.07.2023. URL: <https://www.brainlatam.com/blog/el-cerebro-musical-1199>.
- [13] *Apuntes de la cátedra 16.02 Histología y Anatomía*. Fecha de consulta: 18.07.2023.
- [14] *Lóbulos cerebrales para principiantes*. Fecha de consulta: 18.07.2023. Psyc Lab. URL: <https://www.psycholab.com/lobulos-cerebrales-para-principiantes/>.
- [15] Cinthia Serrano MD. *Corteza prefrontal*. Fecha de consulta: 18.07.2023. Ken Hub. URL: <https://www.kenhub.com/es/library/anatomia-es/corteza-prefrontal>.
- [16] *Áreas y funciones cerebrales IV: Corteza Premotora*. Fecha de consulta: 18.07.2023. NEPSA rehabilitación neurológica en Salamanca. URL: <https://nepsa.es/areas-y-funciones-cerebrales-iv-corteza-premotora/>.

- [17] *Corteza motora primaria: características y funciones*. Fecha de consulta: 18.07.2023. Psicología y Mente. URL: <https://psicologiaymente.com/neurociencias/corteza-motora-primaria>.
- [18] Kiernan A. John. Barr. *El sistema nervioso humano*. 9.^a ed. Wolters Kluwer.
- [19] Richard S. Snell. *Clinical Neuroanatomy*. 7.^a ed. Wolters Kluwer, 2010.
- [20] Purves D. *Neurociencia*. 3.^a ed. Ed. Médica Panamericana, 2008.
- [21] *Enfermedad de Parkinson*. Fecha de consulta: 12.06.2023. Mayo Clinic. URL: <https://www.mayoclinic.org/es/diseases-conditions/parkinsons-disease/symptoms-causes/syc-20376055#>.
- [22] *Study: Parkinson's Disease More Prevalent in Patients with Appendectomies*. Fecha de consulta: 12.06.2023. scinews. URL: <https://www.sci.news/medicine/parkinsons-disease-appendectomy-07173.html>.
- [23] Moustafa AA et al. «Motor symptoms in Parkinson's disease: A unified framework». En: *Neurosci Biobehav Rev*. 68 (2016), págs. 727-740. DOI: <https://doi.org/10.1016/j.neubiorev.2016.07.010>.
- [24] Rosqvist K et al. «Levodopa Effect and Motor Function in Late Stage Parkinson's Disease». En: *J Parkinsons Dis* 8.1 (2018), págs. 59-70. DOI: <https://doi.org/10.3233/JPD-171181>.
- [25] Yamagami M. et al. «Effects of virtual reality environments on overground walking in people with Parkinson disease and freezing of gait». En: *Disability and rehabilitation: Assistive technology* 18.3 (2023), págs. 266-273. DOI: <https://doi.org/10.1080/17483107.2020.1842920>.
- [26] Esperanza Bausela Herreras. «Funciones ejecutivas: nociones del desarrollo desde una perspectiva neuropsicológica». En: *Acción Psicológica* 11.1 (2014). DOI: <https://dx.doi.org/10.5944/ap.1.1.13789>.
- [27] Gonzalez-Latapi P et al. «Cognitive Impairment in Parkinson's Disease: Epidemiology, Clinical Profile, Protective and Risk Factors». En: *Behav Sci (Basel)* 11.5 (2021), pág. 74. DOI: <https://doi.org/10.3390/bs11050074>.
- [28] *Cognitive Changes*. Fecha de consulta: 20.07.2023. Parkinson's Foundation. URL: <https://www.parkinson.org/understanding-parkinsons/non-movement-symptoms/cognitive>.
- [29] Salazar R. D. et al. «Dual Tasking in Parkinson's Disease: Cognitive Consequences While Walking». En: *Neuropsychology* 31.6 (2017), págs. 613-623. DOI: <https://doi.org/10.1037/neu0000331>.
- [30] Raffegau T. E. et al. «A meta-analysis: Parkinson's disease and dual-task walking». En: *Parkinsonism and related disorders* 62 (2019), págs. 28-35. DOI: <https://doi.org/10.1016/j.parkreldis.2018.12.012>.
- [31] Cófreces P et al. «Enfermedad de Parkinson: una actualización bibliográfica de los aspectos psicosociales». En: *Rev Fac Cien Med Univ Nac Cordoba* 79.2 (2022), págs. 181-187. DOI: <https://doi.org/10.31053/1853.0605.v79.n2.33610>.
- [32] progression Parkinsonism: onset y mortality. «Hoehn MM and Yahr MD». En: *Neurology* 17.5 (1967), págs. 427-42. DOI: <https://doi.org/10.1212/wnl.17.5.427>.

- [33] *Parkinson's Disease: Causes, Symptoms, and Treatments*. Fecha de consulta: 18.07.2023. URL: <https://www.nia.nih.gov/health/parkinsons-disease>.
- [34] Rosqvist K et al. «Levodopa Effect and Motor Function in Late Stage Parkinson's Disease». En: *J Parkinsons Dis* 8.1 (2018), págs. 59-70. DOI: <https://doi.org/10.3233/JPD-171181>.
- [35] *Rehabilitación*. Fecha de consulta: 30.07.2023. OPS/OMS. URL: <https://www.paho.org/es/temas/rehabilitacion>.
- [36] Institute for Quality y Efficiency in Health Care. *InformedHealth.org*. 2006.
- [37] Petzinger G. M. et al. «Exercise-enhanced neuroplasticity targeting motor and cognitive circuitry in Parkinson's disease». En: *The Lancet. Neurology* 12.7 (2013), págs. 716-726. DOI: [https://doi.org/10.1016/S1474-4422\(13\)70123-6](https://doi.org/10.1016/S1474-4422(13)70123-6).
- [38] Sehm B. et al. «Structural brain plasticity in Parkinson's disease induced by balance training». En: *Neurobiology of aging* 35.1 (2014), págs. 232-239. DOI: <https://doi.org/10.1016/j.neurobiolaging.2013.06.021>.
- [39] Monteiro-Junior R. S. et al. «We need to move more: Neurobiological hypotheses of physical exercise as a treatment for Parkinson's disease». En: *Medical hypotheses* 85.5 (2015), págs. 537-541. DOI: <https://doi.org/10.1016/j.mehy.2015.07.011>.
- [40] Abbruzzese G. et al. «Rehabilitation for Parkinson's disease: Current outlook and future challenges». En: *Parkinsonism and related disorders* 22.Suppl 1 (2016), S60-S64. DOI: <https://doi.org/10.1016/j.parkreldis.2015.09.005>.
- [41] Protas E. J. et al. «Gait and step training to reduce falls in Parkinson's disease». En: *NeuroRehabilitation* 20.3 (2005), págs. 183-190.
- [42] Mirelman A. et al. «Virtual reality for gait training: can it induce motor learning to enhance complex walking and reduce fall risk in patients with Parkinson's disease?» En: *The journals of gerontology. Series A. Biological sciences and medical sciences* 66.2 (2011), págs. 234-240. DOI: <https://doi.org/10.1093/gerona/glq201>.
- [43] Prof. Dr. Marcelo Merello. *Parkinson. Guía esencial para pacientes, familiares y cuidadores*. Sociedad Iberoamericana de Información Científica (SIIC), 2017. URL: <https://drive.google.com/file/d/1IdNbDpRmP4zbmsmJxGW5c22sXa4euJqN/view>.
- [44] *Parkinson: Ejercicios para todos los días*. Fecha de consulta: 30.07.2023. 2018. URL: <https://www.fleni.org.ar/novedades/parkinson-ejercicios-para-todos-los-dias/>.
- [45] Dr. Beth Templin. *Better Brain Health and Balance*. Fecha de consulta: 31.07.2023. American Parkinson Disease Association. URL: https://www.apdaparkinson.org/wp-content/uploads/2022/05/MayJue22_Newsletterv3.pdf.
- [46] *Cognitive Exercises*. Fecha de consulta: 31.07.2023. American Parkinson Disease Association. URL: https://www.apdaparkinson.org/wp-content/uploads/2022/05/MayJue22_Newsletterv3.pdf.
- [47] *10 ejercicios de estimulación cognitiva para la enfermedad de Parkinson*. Fecha de consulta: 31.07.2023. NeuronUP, 2022. URL: <https://www.neuronup.com/estimulacion-y-rehabilitacion-cognitiva/enfermedades-neurodegenerativas/enfermedad-de-parkinson/10-ejercicios-de-estimulacion-cognitiva-para-la-enfermedad-de-parkinson/>.

- [48] *10 actividades para personas con párkinson durante su tiempo libre*. Fecha de consulta: 31.07.2023. Fundación Degen, 2022. URL: <https://conoceelparkinson.org/cuidados/actividades-personas-parkinson/>.
- [49] *Consejos sobre trastornos cognitivos para paicentes con enfermedad de Parkinson: Cuaderno de ejercicios*. Fecha de consulta: 31.07.2023. Ayuntamiento de Córdoba. URL: <https://ssm.cordoba.es/images/pdf/mayores-ejercicio-memoria-parkison.pdf>.
- [50] Mohd Javaid y Abid Haleem. «Virtual reality applications toward medical field». En: *Clinical Epidemiology and Global Health* 8.2 (2020), págs. 600-605. DOI: <https://doi.org/10.1016/j.cegh.2019.12.010>.
- [51] *What is VR? The devices and apps that turn the real world virtual*. Fecha de consulta: 31.07.2023. NBC News, 2018. URL: <https://www.nbcnews.com/mach/science/what-vr-devices-apps-turn-real-world-virtual-ncna857001>.
- [52] M. I. Berkman y Akan E. «Presence and Immersion in Virtual Reality». En: *Encyclopedia of Computer Graphics and Games* (2018), págs. 1-10. DOI: [10.1007/978-3-319-08234-9_162-1](https://doi.org/10.1007/978-3-319-08234-9_162-1).
- [53] Jean-Christophe Servotte et al. «Virtual Reality Experience: Immersion, Sense of Presence, and Cybersickness». En: *Clinical Simulation in Nursing* 38 (2020), págs. 35-43. ISSN: 1876-1399. DOI: <https://doi.org/10.1016/j.ecns.2019.09.006>. URL: <https://www.sciencedirect.com/science/article/pii/S1876139918303244>.
- [54] Insook Han et al. «Immersive virtual reality for increasing presence and empathy». En: *Journal of Compute Assisted Learning* 38.4 (2022), págs. 1115-1126. DOI: <https://doi.org/10.1111/jcal.12669>.
- [55] *3 Types of Virtual Reality Simulations Used Today*. Fecha de consulta: 27.10.2023. VR Kingdom. URL: <https://vrkingdom.com.au/3-types-of-virtual-reality/>.
- [56] Cremaschi Fabián et al. «Reducción de efectos adversos neurológicos inducidos en entornos de realidad virtual». En: *Revista Médica Universitaria FCM UNCUYO* 16.2 (2020).
- [57] Jose Luis Dorado Collazos. «Un caso de estudio para reducir el cybersickness en el contexto de tecnicas de navegación en sitio y el Oculus Rift». Tesis doct. Universidad de los Andes, 2014. URL: <https://repositorio.uniandes.edu.co/server/api/core/bitstreams/e8eb567a-3dca-409c-9f1f-67ea9829f09c/content>.
- [58] Weech S., Kenny S. y Barnett-Cowan M. «Presence and Cybersickness in Virtual Reality Are Negatively Related: A Review». En: *Frontiers in psychology* 10.158 (2019). DOI: <https://doi.org/10.3389/fpsyg.2019.00158>.
- [59] *Cambiar la frecuencia de actualización de la pantalla en Windows*. Fecha de consulta: 05.08.2023. Microsoft. URL: <https://support.microsoft.com/es-es/windows/cambiar-la-frecuencia-de-actualizaci%C3%B3n-de-la-pantalla-en-windows-c8ea729e-0678-015c-c415-f806f04aae5a>.
- [60] Stanney K.M., Kennedy R.S. y Drexler J.M. «Cybersickness is Not Simulator Sickness». En: *SAGE Publications* 41.2 (1997), págs. 1138-1142. DOI: <https://doi.org/10.1177/107118139704100292>.

- [61] Sevinc V. y Berkman M.I. «Psychometric evaluation of Simulator Sickness Questionnaire and its variants as a measure of cybersickness in consumer virtual environments». En: *Applied ergonomics* 82 (2020). DOI: <https://doi.org/10.1016/j.apergo.2019.102958>.
- [62] Panagiotis Kourtesis et al. «Cybersickness in Virtual Reality Questionnaire (CSQ-VR): A Validation and Comparison against SSQ and VRSQ». En: *Virtual Worlds* 2.1 (2023), págs. 16-35. DOI: <https://doi.org/10.3390/virtualworlds2010002>.
- [63] Riess T. J. «Gait and Parkinson's disease: a conceptual model for an augmented-reality based therapeutic device». En: *Studies in health technology and informatics* 58 (1998), págs. 200-208. URL: <https://pubmed.ncbi.nlm.nih.gov/10350920/>.
- [64] Wenchieh Y et al. «Home-based virtual reality balance training and conventional balance training in Parkinson's disease: A randomized controlled trial». En: *Journal of the Formosan Medical Association = Taiwan yi zhi* 115.9 (2016), págs. 734-743. DOI: <https://doi.org/10.1016/j.jfma.2015.07.012>.
- [65] Shen X. y Mak M. K. «Balance and Gait Training With Augmented Feedback Improves Balance Confidence in People With Parkinson's Disease: A Randomized Controlled Trial». En: *Neurorehabilitation and neural repair* 28.6 (2014), págs. 524-535. DOI: <https://doi.org/10.1177/1545968313517752>.
- [66] Gandolfi M. et al. «Virtual Reality Telerehabilitation for Postural Instability in Parkinson's Disease: A Multicenter, Single-Blind, Randomized, Controlled Trial». En: *BioMed research international* 7962826 (2017). DOI: <https://doi.org/10.1155/2017/7962826>.
- [67] Zhicheng L. et al. «Effects of Virtual Reality Balance Game on Balance Function for Parkinson's Disease». En: *Balkan medical journal* 9 (2016), págs. 1059-1063.
- [68] Özgönenel L. et al. «Use of Game Console for Rehabilitation of Parkinson's Disease». En: *Balkan medical journal* 33.4 (2016), págs. 396-400. DOI: <https://doi.org/10.5152/balkanmedj.2016.16842>.
- [69] Ribas C. G. et al. «Effectiveness of exergaming in improving functional balance, fatigue and quality of life in Parkinson's disease: A pilot randomized controlled trial». En: *Parkinsonism and related disorders* 38 (2017), págs. 13-18. DOI: <https://doi.org/10.1016/j.parkreldis.2017.02.006>.
- [70] Si C. et al. «Effects of Virtual Reality Rehabilitation on Balance for Patients with Parkinson's Disease». En: *Chin J Rehabil Theory Pract* 9 (2017), págs. 1091-1095. DOI: <https://doi.org/10.1371/journal.pone.0224819>.
- [71] Pompeu J. E. et al. «Effect of Nintendo Wii™-based motor and cognitive training on activities of daily living in patients with Parkinson's disease: a randomised clinical trial». En: *Physiotherapy* 98.3 (2012), págs. 196-204. DOI: <https://doi.org/10.1016/j.physio.2012.06.004>.
- [72] Shih M. C. et al. «Effects of a balance-based exergaming intervention using the Kinect sensor on posture stability in individuals with Parkinson's disease: a single-blinded randomized controlled trial». En: *Journal of neuroengineering and rehabilitation* 13.1 (2016), págs. 78. DOI: <https://doi.org/10.1186/s12984-016-0185-y>.

- [73] Alves M. L. M. et al. «Nintendo Wii™ Versus Xbox Kinect™ for Assisting People With Parkinson's Disease». En: *Perceptual and motor skills* 125.3 (2018), págs. 546-565. DOI: <https://doi.org/10.1177/0031512518769204>.
- [74] Liao Y. Y. et al. «Virtual Reality-Based Training to Improve Obstacle-Crossing Performance and Dynamic Balance in Patients With Parkinson's Disease». En: *Neurorehabilitation and neural repair* 29.7 (2015), págs. 658-667. DOI: <https://doi.org/10.1177/1545968314562111>.
- [75] van den Heuvel M. R. et al. «Effects of augmented visual feedback during balance training in Parkinson's disease: a pilot randomized clinical trial». En: *Parkinsonism and related disorders* 20.12 (2014), págs. 1352-1358. DOI: <https://doi.org/10.1016/j.parkreldis.2014.09.022>.
- [76] Ferraz D. D. et al. «The Effects of Functional Training, Bicycle Exercise, and Exergaming on Walking Capacity of Elderly Patients With Parkinson Disease: A Pilot Randomized Controlled Single-blinded Trial». En: *Archives of physical medicine and rehabilitation* 99.5 (2018), págs. 826-833. DOI: <https://doi.org/10.1016/j.apmr.2017.12.014>.
- [77] Pedreira G. et al. «Virtual games and quality of life in Parkinson's disease: A randomised controlled trial». En: *Adv Parkinson's Dis.* 4 (2013), págs. 97-101.
- [78] Lee N. Y., Lee D. K. y Song H. S. «Effect of virtual reality dance exercise on the balance, activities of daily living, and depressive disorder status of Parkinson's disease patients». En: *Journal of physical therapy science* 27.1 (2015), págs. 145-147. DOI: <https://doi.org/10.1589/jpts.27.145>.
- [79] Maggio M. G. et al. «What About the Role of Virtual Reality in Parkinson Disease's Cognitive Rehabilitation? Preliminary Findings From a Randomized Clinical Trial». En: *Journal of geriatric psychiatry and neurology* 31.6 (2018), págs. 312-318. DOI: <https://doi.org/10.1177/0891988718807973>.
- [80] *Realidad Virtual para la rehabilitación, made in Córdoba*. Fecha de consulta: 27.10.2023. Salud en línea. URL: <https://saludenlinea.com.ar/2022/01/31/realidad-virtual-para-la-rehabilitacion-made-in-cordoba/>.
- [81] *Graduados/as en acción: "Creamos una app de rehabilitación a través de juegos de realidad virtual y eso motiva mucho al paciente"*. Fecha de consulta: 27.10.2023. Universidad Nacional Villa María. URL: <https://humanas.unvm.edu.ar/graduados-as-en-accion-creamos-una-app-de-rehabilitacion-a-traves-de-juegos-de-realidad-virtual-y-eso-motiva-mucho-al-paciente/>.
- [82] *Fleni Escobar incorpora nuevas tecnologías aplicadas a la rehabilitación neurológica*. Fecha de consulta: 03.04.2023. Fleni. URL: <https://www.fleni.org.ar/novedades/fleni-escobar-completa-su-proyecto-de-actualizacion-tecnologica/>.
- [83] *C-Mill by Motek*. Fecha de consulta: 15.08.2023. Hocoma. URL: <https://www.hocoma.com/solutions/c-mill/>.
- [84] *Hocoma Motek C-Mill VR+ Dynamic Body Weight Support Treadmill*. Fecha de consulta: 15.08.2023. Rehab Technology. URL: <https://www.rehabtechnology.com.au/product/hocoma-motek-c-mill-vr-dynamic-body-weight-support-treadmill/>.

- [85] *¿Qué es una unidad de medición inercial - IMU?* Fecha de consulta: 27.10.2023. SBG Systems. URL: <https://www.sbg-systems.com/es/unidad-de-medicion-inercial-sensor-imu/>.
- [86] *V3 - Vector Magnitude*. Fecha de consulta: 27.10.2023. vcalc. URL: <https://www.vcalc.com/wiki/vector-magnitude>.
- [87] Fernando Kabas. «Sistema de captura de parámetros espacio-temporales para laboratorio de marcha basado en sensores MEMS». Tesis doct. Instituto Tecnológico de Buenos Aires (ITBA), 2020.
- [88] Wandinger David et al. «Multibody Dynamics of a Serial Robot With Four Axes». En: (2021). URL: https://www.researchgate.net/publication/362508568_Multibody_Dynamics_of_a_Serial_Robot_With_Four_Axes.
- [89] *Unidad de Medida Inercial. Algoritmo de Estimación e Implementación Software. Capítulo 2 - Desarrollo teórico*. Fecha de consulta: 28.10.2023. Universidad de Sevilla. URL: https://biblus.us.es/bibing/proyectos/abreproy/11253/fichero/memoria%252F02_capitulo2.pdf.
- [90] *HP Reverb G2 manual*. Fecha de consulta: 16.05.2023. HP. URL: <https://www.manual.ar/hp/reverb-g2/manual?p=5>.
- [91] *Medida estándar de un living*. Fecha de consulta: 16.05.2023. Planos de Casas Gratis. URL: <https://planosycasas.net/medida-estandar-de-un-living/>.
- [92] Fecha de consulta: 14.07.2023. URL: <https://computerrentals.be/product/hp-reverb-2-headset-with-controllers/>.
- [93] Fecha de consulta: 14.07.2023. URL: https://store.steampowered.com/app/1271440/HP_Reverb_G2/?l=spanish.
- [94] *How do you set up the HP Reverb G2?* Fecha de consulta: 05.09.2023. coolblue. URL: <https://www.coolblue.nl/en/advice/how-do-you-set-up-the-hp-reverb-g2.html>.
- [95] *Unity: ¿Qué es y cómo funciona?* Fecha de consulta: 16.08.2023. Unity. URL: <https://support.unity.com/hc/es/articles/7642130833812-Unity-Qu%C3%A9-es-y-c%C3%B3mo-funciona-#:~:>.
- [96] *Integrated development environment (IDE) support*. Fecha de consulta: 16.08.2023. Unity. URL: <https://docs.unity3d.com/Manual/ScriptingToolsIDEs.html>.
- [97] *PROGRAMAS PARA CREAR REALIDAD VIRTUAL*. Fecha de consulta: 17.05.2023. IAMVR. URL: <https://i-amvr.com/programas-para-crear-realidad-virtual/>.
- [98] *Pros y contras de programar con Unity vs. Unreal Engine*. Fecha de consulta: 17.05.2023. Deusto Formación. URL: <https://www.deustoformacion.com/blog/disenio-produccion-audiovisual/pros-contras-programar-unity-vs-unreal-engine>.
- [99] *SALUD Y BIENESTAR EN LA ERA DIGITAL*. Fecha de consulta: 17.05.2023. Unity. URL: <https://unity.com/es/humanity/showcase/digital-health-and-wellbeing>.
- [100] *Flujo de trabajo de los Assets (Asset Workflow)*. Fecha de consulta: 16.08.2023. Unity. URL: <https://docs.unity3d.com/es/530/Manual/AssetWorkflow.html>.
- [101] *Unity User Manual*. Fecha de consulta: 17.08.2023. Unity. URL: <https://docs.unity3d.com/es/2018.4/Manual/UnityManual.html>.

- [102] *Lucidchart*. Fecha de consulta: 30.09.2023. Lucid. URL: https://www.lucidchart.com/pages/landing?utm_source=google&utm_medium=cpc&utm_campaign=_chart_en_tier3_mixed_search_brand_exact_&km_CPC_CampaignId=1484560207&km_CPC_AdGroupId=60168114191&km_CPC_Keyword=lucid%20chart&km_CPC_MatchType=e&km_CPC_ExtensionID=&km_CPC_Network=g&km_CPC_AdPosition=&km_CPC_Creative=442433234360&km_CPC_TargetID=kwd-55720648523&km_CPC_Country=1000073&km_CPC_Device=c&km_CPC_placement=&km_CPC_target=&gclid=CjwKCAjw9-6oBhBaEiwAHv1QvNy-Uvb1fFlSq6lFAt4clx1uRhFMTh-J9Gm6GfR4C-bXA3-c9fc4Nx0CALQQA_vD_BwE.
- [103] Shu Y et al. «Impacts of Landscape Type, Viewing Distance, and Permeability on Anxiety, Depression, and Stress». En: *Int J Environ Res Public Health* 19.16 (2022), pág. 9867. DOI: <https://doi.org/10.3390/ijerph19169867>.
- [104] Unity Technologies. *Terrain Sample Asset Pack*. Fecha de compra: 31.03.2023. URL: <https://assetstore.unity.com/packages/3d/environments/landscapes/terrain-sample-asset-pack-145808>.
- [105] forst. *Conifers [BOTD]*. Fecha de compra: 11.04.2023. URL: <https://assetstore.unity.com/packages/3d/vegetation/trees/conifers-botd-142076>.
- [106] Vladislav Pochezhertsev. *Grass And Flowers Pack 1*. Fecha de compra: 11.04.2023. URL: <https://assetstore.unity.com/packages/2d/textures-materials/nature/grass-and-flowers-pack-1-17100>.
- [107] A dog's life software. *Outdoor Ground Textures*. Fecha de compra: 11.04.2023. URL: <https://assetstore.unity.com/packages/2d/textures-materials/floors/outdoor-ground-textures-12555>.
- [108] rpgwhitelock. *AllSky Free - 10 Sky / Skybox Set*. Fecha de compra: 05.05.2023. URL: <https://assetstore.unity.com/packages/2d/textures-materials/sky/allsky-free-10-sky-skybox-set-146014>.
- [109] ¿Qué es la temperatura de color? Fecha de consulta: 21.07.2023. Celer Light. URL: <https://www.celerlight.com/que-es-la-temperatura-de-color/>.
- [110] alkakrab. *25 Fantasy RPG Game Tracks Music Pack*. Fecha de compra: 02.05.2023. URL: <https://assetstore.unity.com/packages/audio/music/25-fantasy-rpg-game-tracks-music-pack-240154>.
- [111] NoxSound. *Nature - Essentials*. Fecha de compra: 03.07.2023. URL: <https://assetstore.unity.com/packages/audio/ambient/nature/nature-essentials-208227>.
- [112] Dustyroom. *FREE Casual Game SFX Pack*. Fecha de compra: 02.03.2023. URL: <https://assetstore.unity.com/packages/audio/sound-fx/free-casual-game-sfx-pack-54116>.
- [113] *Young-Onset Parkinson's Disease*. Fecha de consulta: 05.09.2023. Johns Hopkins Medicine. URL: <https://www.hopkinsmedicine.org/health/conditions-and-diseases/parkinsons-disease/youngonset-parkinsons-disease>.
- [114] *Parkinson's Disease: Causes, Symptoms, and Treatments*. Fecha de consulta: 05.09.2023. National Institute on Aging. URL: <https://www.nia.nih.gov/health/parkinsons-disease>.

- [115] *SUS - Escala de usabilidad del sistema*. Fecha de consulta: 05.09.2023. Redesign Reality. URL: <https://you-x.eu/es/escala-de-usabilidad-del-sistema-sus/>.
- [116] *System Usability Scale (SUS)*. Fecha de consulta: 05.09.2023. usability gov. URL: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>.
- [117] Jeff Sauro PhD. *5 Ways to Interpret a SUS Score*. Fecha de consulta: 10.10.2023. MeasuringU. URL: <https://measuringu.com/interpret-sus-score/>.
- [118] *¿Qué es Net Promoter Score (NPS)®? Definición y ejemplos*. Fecha de consulta: 10.10.2023. qualtrics. URL: <https://www.qualtrics.com/es-la/gestion-de-la-experiencia/cliente/net-promoter-score/>.
- [119] Panagiotis Kourtesis et al. «CyberSickness in Virtual Reality Questionnaire (CSQ-VR): A brief tool for evaluating the Virtual Reality Induced Symptoms and Effects (VRISE)». En: (2022). DOI: <http://dx.doi.org/10.13140/RG.2.2.36571.03362>.
- [120] *Apuntes de la cátedra 16.24 Ingeniería de Rehabilitación*. Fecha de consulta: 30.07.2023.
- [121] *Xsens DOT User Manual*. Fecha de consulta: 05.10.2023. Xsens Technologies B.V. URL: <https://www.xsens.com/hubfs/Downloads/DOT/Documents/2021-07%20-%20Archived%20-%20Xsens%20DOT%20User%20Manual%20.pdf>.
- [122] *El paso humano*. Fecha de consulta: 04.11.2023. CASIO Educación España. URL: <https://www.edu-casio.es/wp-content/uploads/2020/04/El-paso-humano.pdf>.
- [123] Hernán Marty. *¿Cuánto mide una cuadra?* Fecha de consulta: 08.11.2023. Cronos. URL: <https://infocronos.com.ar/nota/14816/cuanto-mide-una-cuadra/>.

8. Glosario

2D segunda dimensión. 20, 33

3D tercera dimensión. 20, 33, 34, 76

CDM congelación de la marcha. 15, 17, 25

CETtar Centro de Evaluación y Tratamiento con Tecnologías Aplicadas a la Rehabilitación. 27

CSQ-VR Cuestionario de Cibermareo en Realidad Virtual. 24, 48, 50, 51, 61–63, 72, 73, 76, 116–119

DT dual-tasking. 16, 19, 24

ECG electrocardiograma. 54

ECP Estimulación Cerebral Profunda. 6, 18

ENU East North Up. 51

- EP** enfermedad de Parkinson. 2, 6, 7, 9, 11, 13–21, 24, 26, 28, 31, 39, 47, 76
- Fleni** Fundación para la Lucha contra las Enfermedades Neurológicas de la Infancia. 20, 27, 28, 51
- GPE** globo pálido externo. 13, 14
- GPI** globo pálido interno. 13, 14
- IDE** Entorno de Desarrollo Integrado. 33, 35
- IMU** Unidad de Medición Inercial. 29, 30, 51
- ITBA** Instituto Tecnológico de Buenos Aires. 46
- LTS** Long Term Support. 34
- NPS** Net Promoter Score. 49, 50, 59, 72
- OMS** Organización Mundial de Salud. 6
- RA** realidad aumentada. 19, 24, 28, 35
- RV** realidad virtual. 2, 6, 7, 19, 21–28, 31–33, 35, 39, 47, 48, 50, 71–73, 76, 77
- SN** sistema nervioso. 8, 16, 17
- SNC** sistema nervioso central. 8, 9, 11, 12
- SNP** sistema nervioso periférico. 8
- SSQ** Cuestionario de malestar asociado al simulador. 24, 73
- SUS** Sistema de Escala de Usabilidad. 48–50, 59–61, 71, 72, 76, 113–115
- TT** treadmill training. 19, 24
- VA/VL** ventral anterior y ventral lateral. 13, 14
- VRISE** Síntomas y Efectos Inducidos por la Realidad Virtual. 23, 24, 50
- VRNQ** Cuestionario de Neurociencia de Realidad Virtual. 50
- VRSQ** Cuestionario de malestar asociado a realidad virtual. 24
- XKFCore** Xsens Kalman Filter core. 51

9. Anexo

9.1. *Scripts* del programa en Unity

En esta sección del Anexo, se muestran los *scripts* utilizados en el programa de Unity para diseñar la aplicación *BrainWalkVR*, y sus funciones.

N°	Nombre del <i>script</i>	Función
1	<i>AnimateHandOnInput</i>	Controlar la animación de un modelo 3D de una mano virtual, en respuesta a la entrada del usuario (<i>trigger</i> y <i>grab</i>).
2	<i>BackgroundMusicManager</i>	Controlar la reproducción de dos pistas de sonido: la música de fondo en común para todos los escenarios y la música de ambiente según el escenario seleccionado.
3	<i>DetectText</i>	Administrar el juego de palabras en <i>BrainWalkVR</i> : proporciona la lógica de juego, la generación de piezas de letras, la gestión de la interfaz de usuario, y la transición entre escenas al final del juego.
4	<i>DontDestroyCategory</i>	Guardar la categoría de palabras que haya elegido el usuario en la escena de inicio, para que aparezcan palabras de esa categoría en la escena principal.
5	<i>EndMenu</i>	Administrar las opciones del menú en la escena final: permite salir del juego o reiniciarlo después de haber terminado una sesión de entrenamiento
6	<i>FadeScreen</i>	Brindar la pantalla que desvanece (<i>fade screen</i>) cuando inicia o finaliza una escena, para que la transición no sea abrupta.
7	<i>GameMenuManager</i>	Controlar la visibilidad y la posición de la interfaz de usuario durante el juego.
8	<i>MoveWithHand</i>	Permite que una pieza de letra se pueda agarrar y soltar por el usuario. Devuelve un sonido como feedback cuando una pieza es agarrada.
9	<i>PuzzlePieceSpawner</i>	Gestionar la generación de piezas de letras en función de la categoría seleccionada: garantiza que las piezas se generen en posiciones válidas y aleatorias en el espacio del entrenamiento.
10	<i>SceneTransitionManager</i>	Gestionar las transiciones entre escenas del juego.
11	<i>SetBackgroundVolume</i>	Permitir al usuario ajustar el volumen de dos pistas de sonido en el fondo, usando los sliders en la interfaz de usuario.
12	<i>SetNumberOfRevealedLetters</i>	Permitir al usuario elegir el número de letras ya reveladas al empezar a completar una palabra en el juego, usando un slider, para controlar el nivel de dificultad del entrenamiento.

N°	Nombre del <i>script</i>	Función
13	<i>SetSoundVolume</i>	Permitir al usuario ajustar el volumen de los sonidos de feedback a las acciones del usuario.
14	<i>ToggleGroupSelectionManager</i>	Gestionar la selección de opciones en <i>toggle groups</i> en el menú de inicio, donde el usuario debe elegir la categoría de palabras, el nivel de dificultad y el escenario para el juego.
15	<i>ToggleSystem</i>	Garantizar que los <i>toggle groups</i> permitan seleccionar una sola opción.

Cuadro 6: Nombres de los *scripts*

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.InputSystem;
5
6 public class AnimateHandOnInput : MonoBehaviour
7 {
8     public InputActionProperty pinchAnimationAction;
9     public InputActionProperty gripAnimationAction;
10    public Animator handAnimator;
11
12    void Update()
13    {
14        float triggerValue = pinchAnimationAction.action.ReadValue<float>();
15        handAnimator.SetFloat("Trigger", triggerValue);
16
17        float gripValue = gripAnimationAction.action.ReadValue<float>();
18        handAnimator.SetFloat("Grip", gripValue);
19    }
20 }

```

Listing 1: *AnimateHandOnInput*

```

1 using UnityEngine;
2
3 public class BackgroundMusicManager : MonoBehaviour
4 {
5     public AudioSource natureSound;
6     public AudioSource musicSound;
7
8     void Start()
9     {
10        natureSound.Play();
11        Invoke("PlayMusic", 5);
12    }
13
14    void PlayMusic()
15    {
16        musicSound.Play();
17    }
18 }

```


19 }

Listing 2: *BackgroundMusicManager*

```

1 using System.Collections.Generic;
2 using UnityEngine;
3 using TMPro;
4 using System.Text;
5 using UnityEngine.SceneManagement;
6 using System.Linq;
7 using System.Collections;
8
9 public class DetectText : MonoBehaviour
10 {
11     // Text
12     public TextMeshProUGUI wordText; // String con guion bajo
13     public TextMeshProUGUI answerText; // String de respuesta que esta escondido
14     public TextMeshProUGUI errorText; // Cantidad de errores
15     public TextMeshProUGUI roundText; // Numero de ronda
16     public TextMeshProUGUI encabezadoTema;
17     public TextMeshProUGUI textoTema;
18     public TextMeshProUGUI encabezadoError;
19     public TextMeshProUGUI textoVictoria;
20     public TextMeshProUGUI textoErrorFinal;
21     public TextMeshProUGUI encabezadoErrorFinal;
22
23     private string hiddenWord; // answerText.text
24     private string underscoreString; // wordText.text
25     private string errorString; // errorText.text
26     private string roundString; // roundText.text
27
28     // Audio
29     public AudioSource correctSound;
30     public AudioSource incorrectSound;
31     public AudioSource completeSound;
32     public AudioSource victorySound; // Sonido de victoria cuando termina de
33     // completar 6 rondas
34     public int nextRound = 1;
35     public int error = 0;
36     public int roundNumber;
37
38     public List<GameObject> victoryScreenObjects; // Lista de objetos que aparecen
39     // en la pantalla cuando terminan las 6 rondas
40
41     private void OnCollisionEnter(Collision collision)
42     {
43         TextMeshPro textMeshPro = collision.gameObject.GetComponentInChildren<
44         TextMeshPro>(); // Chequea si el objeto colisionado tiene un componente
45         TextMeshPro
46         hiddenWord = answerText.text;
47         underscoreString = wordText.text;
48         errorString = errorText.text;
49
50         if (textMeshPro != null) // Si el objeto colisionado tiene un componente
51         TextMeshPro
52         {
53             string letter = textMeshPro.text; // La letra de la pieza colisionada

```

```

49     con la mesa
50         int cantidadLetras = hiddenWord.Count(c => c == letter[0]);
51         int cantidadLetrasReveladas = underscoreString.Count(c => c == letter
52 [0]);
53
54         if (underscoreString.Contains(letter) && cantidadLetras ==
55 cantidadLetrasReveladas) // Si la letra ya fue revelada antes
56         {
57             Debug.Log("Letter already revealed: " + letter);
58             error ++;
59             errorText.text = error.ToString();
60             incorrectSound.Play();
61             Destroy(collision.gameObject); // Destruye la pieza colisionada con
62 la mesa
63         }
64         else // Si la letra no fue revelada antes
65         {
66             bool letterFound = hiddenWord.Contains(letter); // Chequea si la
67 letra pertenece a la palabra
68
69             if (letterFound) // Si la letra pertenece a la palabra
70             {
71                 // Actualiza la cadena de guiones bajos con la letra revelada
72                 StringBuilder builder = new StringBuilder(underscoreString);
73                 for (int i = 0; i < hiddenWord.Length; i++)
74                 {
75                     if (hiddenWord[i] == letter[0])
76                     {
77                         builder[i * 2] = letter[0];
78                     }
79                 }
80                 underscoreString = builder.ToString(); // Actualiza la cadena
81 de guiones
82                 wordText.text = underscoreString; // Actualiza la cadena de
83 guiones en el componente TextMeshProUGUI de wordText en Game Screen
84
85                 Debug.Log("Letter found: " + letter);
86
87                 string convertedUnderscoreString =
88 ConvertUnderscoreStringToWord(underscoreString);
89
90                 if (convertedUnderscoreString == hiddenWord) // Si la palabra
91 ya est completa
92                 {
93                     Destroy(collision.gameObject);
94                     Debug.Log("Word completed!");
95                     completeSound.Play();
96                     Invoke("ExecuteOnRoundChanged", 3); // Incrementa el numero
97 de ronda despues de 3 segundos
98                 }
99                 else
100                 {
101                     correctSound.Play();
102                     Destroy(collision.gameObject);
103                 }
104             }
105         }

```

```

96         else
97         {
98             incorrectSound.Play();
99             error++;
100             errorText.text = error.ToString(); // Actualiza el texto de
error en Game Screen
101             Debug.Log("Wrong: " + letter);
102             Destroy(collision.gameObject);
103         }
104     }
105 }
106 }
107
108 void ExecuteOnRoundChanged()
109 {
110     nextRound = IncrementRoundCounter(); // Incrementa el numero de ronda
111
112     if (nextRound != 0)
113     {
114         roundText.text = nextRound.ToString();
115         OnRoundChanged(nextRound);
116     }
117 }
118
119 private string ConvertUnderscoreStringToWord(string underscoreString)
120 {
121     string word = underscoreString.Replace("_", "").Replace(" ", "");
122     return word;
123 }
124
125 private int IncrementRoundCounter() // Incrementa el numero de ronda
126 {
127     roundString = roundText.text;
128     int round = int.Parse(roundString);
129     round++;
130
131     if (round > roundNumber) // Cuando se completan las 6 rondas, el juego
termina
132     {
133         victorySound.Play();
134         round = 0;
135
136         textoErrorFinal.text = error.ToString();
137
138         GameObject[] gameScreenObjects = GameObject.FindGameObjectsWithTag("
GameScreen");
139
140         foreach (GameObject gameScreenObject in gameScreenObjects)
141         {
142             gameScreenObject.SetActive(!gameScreenObject.activeSelf); //
Desactiva todos los objetos con el tag "GameScreen"
143         }
144
145         // The Victory Screen appears
146         foreach (GameObject victoryScreenObject in victoryScreenObjects)
147         {
148             victoryScreenObject.SetActive(!victoryScreenObject.activeSelf); //

```

```

149     Activa todos los objetos con el tag "VictoryScreen"
150     }
151
152     // La escena final se carga despues de 7 segundos
153     Invoke("LoadNextScene", 7);
154
155 }
156
157 return round;
158 }
159 public FadeScreen fadeScreen;
160 public void LoadNextScene()
161 {
162     StartCoroutine(GoToSceneAsyncRoutine(4));
163 }
164 IEnumerator GoToSceneAsyncRoutine(int sceneIndex)
165 {
166     fadeScreen.FadeOut();
167     // Launch the new scene
168     AsyncOperation operation = SceneManager.LoadSceneAsync(sceneIndex);
169     operation.allowSceneActivation = false;
170
171     float timer = 0;
172     while (timer <= fadeScreen.fadeDuration && !operation.isDone)
173     {
174         timer += Time.deltaTime;
175         yield return null;
176     }
177
178     operation.allowSceneActivation = true;
179 }
180
181 private List<string> deportesWords = new List<string>() { "FUTBOL", "TENIS", "
HOCKEY", "NATACION", "BOXEO", "PINGPONG" };
182 private List<string> frutasWords = new List<string>() { "MANZANA", "BANANA", "
NARANJA", "MANDARINA", "SANDIA", "FRUTILLA" };
183 private List<string> verdurasWords = new List<string>() { "ZANAHORIA", "
ESPINACA", "PEPINO", "TOMATE", "BERENJENA", "LECHUGA" };
184 private List<string> animalesWords = new List<string>() { "ELEFANTE", "CABALLO"
, "GALLINA", "CONEJO", "PERRO", "CAMELLO" };
185 private TextMeshProUGUI numberOfRevealedLettersText; // Lo puede elegir el
usuario en UI
186
187 private void OnRoundChanged(int round)
188 {
189     string selectedCategory = PlayerPrefs.GetString("Category"); // Obtiene el
texto de la categor a seleccionada
190
191     List<string> selectedWords = GetWordsByCategory(selectedCategory); //
Obtiene la lista de palabras de la categor a seleccionada
192     Debug.Log("Selected category: " + selectedCategory);
193     Debug.Log("Selected words: " + selectedWords.Count);
194     Debug.Log("Round: " + round);
195     string selectedWord = selectedWords[round-1]; // Obtiene la primera palabra
de la lista de palabras de la categor a seleccionada
196     Debug.Log("Selected word: " + selectedWord);
197     int numberOfRevealedLetters = PlayerPrefs.GetInt("Dificultad");

```

```

197     string underscoreString = GenerateUnderscoreString(selectedWord,
numberOfRevealedLetters); // Genera una cadena de guiones bajos de la misma
longitud que la palabra aleatoria
198     wordText.text = underscoreString; // Asigna la cadena de guiones bajos al
componente TextMeshProUGUI de wordText en Game Screen
199
200     SpawnPieces(selectedWord); // Genera las piezas del puzzle
201 }
202
203 private List<string> GetWordsByCategory(string category) // Toma como
par metro una categoria y devuelve su lista de palabras
204 {
205     switch (category)
206     {
207         case "Deportes":
208             return deportesWords;
209         case "Frutas":
210             return frutasWords;
211         case "Verduras":
212             return verdurasWords;
213         case "Animales":
214             return animalesWords;
215         default:
216             return new List<string>();
217     }
218 }
219 private string GenerateUnderscoreString(string word, int
numberOfRevealedLetters)
220 // Toma como par metros una palabra y la cantidad de letras reveladas
221 // Devuelve una cadena de guiones bajos de la misma longitud
222 {
223     string underscoreString = ""; // Cadena vacia para concatenar guiones
bajos con letras reveladas
224     int wordLength = word.Length;
225     List<int> revealedLettersIndexes = new List<int>(); // Lista de indices de
letras reveladas
226
227     for (int j = 0; j < numberOfRevealedLetters; j++)
228     {
229         int randomIndex = Random.Range(0, wordLength); // Genera un indice
aleatorio entre 0 y el tamaño de la palabra
230         if (!revealedLettersIndexes.Contains(randomIndex)) // Si la lista de
indices de letras reveladas no contiene el indice aleatorio, lo agrega
231         {
232             revealedLettersIndexes.Add(randomIndex);
233         }
234         else // Si la lista de indices de letras reveladas ya contiene el
indice aleatorio, genera otro indice aleatorio
235         {
236             j--; // Decrementa el contador para que se vuelva a ejecutar el
ciclo
237         }
238     }
239
240     for (int i = 0; i < wordLength; i++)
241     {
242         if (!revealedLettersIndexes.Contains(i)) // Si la lista de indices de

```

```

243     letras reveladas no contiene el ndice actual, agrega un guion bajo
244     {
245         underscoreString += "_";
246     }
247     else
248     {
249         underscoreString += word[i]; // Si la lista de ndices de letras
250         reveladas contiene el ndice actual, agrega la letra correspondiente
251     }
252     if (i < wordLength - 1) // Si no es la ltima letra, agrega un espacio
253     {
254         underscoreString += " ";
255     }
256 }
257 return underscoreString;
258 }
259
260 // PuzzlePieceSpawner
261 public GameObject piecePrefab;
262 public int numberOfPieces;
263 public GameObject spawnZone1;
264 public GameObject spawnZone2;
265 public GameObject spawnZone3;
266 public float heightY;
267 public List<Vector3> spawnedPositionsZone1 = new List<Vector3>(); // Es una
268 lista vaca al principio
269 public List<Vector3> spawnedPositionsZone2 = new List<Vector3>();
270 public List<Vector3> spawnedPositionsZone3 = new List<Vector3>();
271 public float minDistance; // Distancia minima entre piezas generadas
272 private List<GameObject> spawnedPieces = new List<GameObject>(); // Lista de
273 piezas generadas
274 private int spawnZoneCounter = 1;
275
276 public void SpawnPieces(string word) // Genera las piezas del puzzle
277 {
278     answerText.text = word; // Es la respuesta correcta, est escondida en
279     Game Scene
280
281     // Destruye todos los GameObjects que tenga nombre "Pieza(Clone)"
282     GameObject[] pieces = GameObject.FindGameObjectsWithTag("PieceClone");
283     foreach (GameObject piece in pieces)
284     {
285         Destroy(piece);
286     }
287
288     // Genera nuevas piezas
289     int wordIndex = 0; // indice para recorrer las letras de la palabra
290     for (int i = 0; i < numberOfPieces; i++) // Genera el nmero de piezas
291     especificado en el inspector
292     {
293         GameObject newPiece = Instantiate(piecePrefab); // Crea una pieza copia
294         de la pieza prefab
295         newPiece.transform.position = GetRandomSpawnPosition()[0]; // Se le
296         asigna una posici n aleatoria dentro de una de las 3 zonas de spawn
297         newPiece.transform.rotation = Quaternion.Euler(GetRandomSpawnPosition())

```

```

292 [1]); // Se le asigna una rotacion aleatoria dentro de una de las 3 zonas de
293 spawn
294
295 // Si quedan letras en la palabra, asignar la siguiente letra
296 if (wordIndex < word.Length)
297 {
298     newPiece.GetComponentInChildren<TMPPro.TextMeshPro>().text = word[
299 wordIndex].ToString();
300     wordIndex++;
301 }
302 else
303 {
304     // Si no quedan letras en la palabra, generar una letra aleatoria
305     string randomLetter = ((char)Random.Range(65, 91)).ToString();
306     newPiece.GetComponentInChildren<TMPPro.TextMeshPro>().text =
307 randomLetter;
308 }
309
310 // Asigna el tag "PieceClone" a la pieza generada
311 string cloneTag = "PieceClone";
312 newPiece.tag = cloneTag;
313
314 spawnedPieces.Add(newPiece);
315 }
316 }
317
318 private List<Vector3> GetRandomSpawnPosition() // Genera una posici n
319 aleatoria dentro de una de las 3 zonas de spawn
320 {
321     Bounds bounds; // Bounds = Boundaries of a collider
322     List<Vector3> listaVectores = new List<Vector3>(); // Lista de vectores que
323 va a devolver la funci n
324
325 switch (spawnZoneCounter) // Este contador va a ir aumentando de 1 a 3
326 {
327     case 1:
328         bounds = spawnZone1.GetComponent<BoxCollider>().bounds; // Obtiene
329 los 1 mites del box collider de la zona 1
330         listaVectores = GetValidSpawnPosition(bounds, spawnedPositionsZone1
331 , spawnZoneCounter); // Devuelve una posici n dentro de los 1 mites de la
332 zona 1
333         break;
334     case 2:
335         bounds = spawnZone2.GetComponent<BoxCollider>().bounds; // Obtiene
336 los 1 mites del box collider de la zona 2
337         listaVectores = GetValidSpawnPosition(bounds, spawnedPositionsZone2
338 , spawnZoneCounter); // Devuelve una posici n dentro de los 1 mites de la
339 zona 2
340         break;
341     case 3:
342         bounds = spawnZone3.GetComponent<BoxCollider>().bounds; // Obtiene
343 los 1 mites del box collider de la zona 3
344         listaVectores = GetValidSpawnPosition(bounds, spawnedPositionsZone3
345 , spawnZoneCounter); // Devuelve una posici n dentro de los 1 mites de la
346 zona 3
347         break;
348     default:

```



```

334         bounds = spawnZone1.GetComponent<BoxCollider>().bounds;
335         listaVectores = GetValidSpawnPosition(bounds,
spawnedPositionsZone1, spawnZoneCounter);
336         break;
337     }
338
339     return listaVectores;
340 }
341
342 private List<Vector3> GetValidSpawnPosition(Bounds bounds, List<Vector3>
spawnedPositions, int index)
343 // Toma como par metros los l mites de una zona de spawn y la lista de
posiciones ya generadas en esa zona
344 {
345     do
346     {
347         float x = 0.1f;
348         float z = 0.1f;
349         Vector3 rotationVector = new Vector3(0, 0, 0);
350         List<Vector3> listVectors = new List<Vector3>();
351         if (index == 1)
352         {
353             z = 0.73f;
354             x = Random.Range(bounds.min.x, bounds.max.x);
355             rotationVector = new Vector3(0, -180, 0);
356         }
357         else if (index == 2)
358         {
359             z = -2.77f;
360             x = Random.Range(bounds.min.x, bounds.max.x);
361             rotationVector = new Vector3(0, -270, 0);
362         }
363         else if (index == 3)
364         {
365             x = -4.79f;
366             z = Random.Range(bounds.min.z, bounds.max.z);
367             rotationVector = new Vector3(0, -90, 0);
368         }
369
370         Vector3 spawnPosition = new Vector3(x, heightY, z); // Crea un vector
con la posici n generada
371
372         // Verificar si la nueva posicion esta lo suficientemente lejos de las
posiciones ya generadas
373         bool isValidPosition = true;
374
375         foreach (Vector3 position in spawnedPositions)
376         {
377             // Eval a si la distancia entre la posici n nueva y una posici n
ya generada es menor a la distancia m nima
378             if (Vector3.Distance(spawnPosition, position) < minDistance)
379             {
380                 isValidPosition = false; // Si la distancia es menor a la
distancia m nima, la posici n no es v lida
381                 break; // Sale del ciclo for
382             }
383         }

```

```

384
385         if (isValidPosition) // Si la posici n es v lida , la devuelve y
termina el ciclo do-while
386         {
387             RaycastHit hit; // Raycast = Rayo que sale de un punto en una
direcci n
388             if (Physics.Raycast(new Vector3(x, bounds.max.y + 0.1f, z), -
Vector3.up, out hit, Mathf.Infinity))
389                 // Si el rayo que sale de la posici n generada hacia abajo choca
con algo, la posici n es v lida
390                 // Esto es porque la posici n generada est en el aire, y
queremos que est sobre el suelo
391                 {
392                     spawnPosition = hit.point + (Vector3.up * heightY); // La
posici n generada se ajusta para que est sobre el suelo
393                 }
394
395                 spawnedPositions.Add(spawnPosition); // Se agrega a la lista de
spawn positions
396                 IncrementSpawnZoneCounter(); // Se incrementa el contador de zonas
de spawn
397                 listVectors.Add(spawnPosition);
398                 listVectors.Add(rotationVector);
399                 return listVectors;
400             }
401
402         } while (true); // Este ciclo se va a repetir hasta que se encuentre una
posici n v lida
403     }
404
405     private void IncrementSpawnZoneCounter() // Incrementa el contador de zonas de
spawn
406     {
407         spawnZoneCounter++;
408         if (spawnZoneCounter > 3) // Si ya lleg a generar una pieza por cada
spawn zone, vuelve a empezar desde la zona 1
409         {
410             spawnZoneCounter = 1;
411         }
412         // Es decir, se va llenando 1-2-3, 1-2-3, 1-2-3, ...
413     }
414
415 }

```

Listing 3: *DetectText*

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class DontDestroyCategory : MonoBehaviour
6 {
7     void Awake()
8     {
9         DontDestroyOnLoad(transform.gameObject);
10    }

```

11 }

Listing 4: *DontDestroyCategory*

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class EndMenu : MonoBehaviour
7 {
8     public void QuitGame()
9     {
10         // Salir del juego
11         #if UNITY_EDITOR
12             UnityEditor.EditorApplication.isPlaying = false;
13         #else
14             Application.Quit();
15         #endif
16     }
17
18     public void RestartGame()
19     {
20         // Load the second scene by its build index
21         SceneManager.LoadScene(0);
22     }
23 }

```

Listing 5: *EndMenu*

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class FadeScreen : MonoBehaviour
6 {
7     public bool fadeOnStart = true;
8     public float fadeDuration = 2;
9     public Color fadeColor;
10    private Renderer rend;
11
12    void Start()
13    {
14        rend = GetComponent<Renderer>();
15        if (fadeOnStart)
16        {
17            FadeIn();
18        }
19    }
20
21    public void FadeIn()
22    {
23        Fade(1, 0);
24    }
25
26    public void FadeOut()
27    {

```

```

28     Fade(0, 1);
29 }
30
31 public void Fade(float alphaI, float alphaOut)
32 {
33     StartCoroutine(FadeRoutine(alphaI, alphaOut));
34 }
35
36 public IEnumerator FadeRoutine(float alphaI, float alphaOut)
37 {
38     float t = 0; // timer
39     while (t <= fadeDuration)
40     {
41         Color newColor = fadeColor;
42         newColor.a = Mathf.Lerp(alphaI, alphaOut, t / fadeDuration);
43         rend.material.SetColor("_Color", newColor);
44         t += Time.deltaTime;
45         yield return null; // wait for a frame then continue
46     }
47     Color newColor2 = fadeColor;
48     newColor2.a = alphaOut;
49     rend.material.SetColor("_Color", newColor2);
50 }
51 }

```

Listing 6: *FadeScreen*

```

1 using UnityEngine;
2 using UnityEngine.InputSystem;
3
4 public class GameManager : MonoBehaviour
5 {
6     public Transform head;
7     public float spawnDistance = 2;
8     public GameObject menu;
9     public GameObject rightHandRay;
10    public GameObject leftHandRay;
11    public InputActionProperty showButton;
12    // Start is called before the first frame update
13    void Start()
14    {
15
16    }
17
18    // Update is called once per frame
19    void Update()
20    {
21        if (showButton.action.WasPressedThisFrame())
22        {
23            menu.SetActive(!menu.activeSelf);
24            rightHandRay.SetActive(!rightHandRay.activeSelf);
25            leftHandRay.SetActive(!leftHandRay.activeSelf);
26            menu.transform.position = head.position + new Vector3(head.forward.x,
27            0, head.forward.z).normalized * spawnDistance;
28        }
29
30        menu.transform.LookAt(new Vector3(head.position.x, menu.transform.position.

```

```

    y, head.position.z));
30     menu.transform.forward *= -1;
31 }
32 }

```

Listing 7: *GameMenuManager*

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.XR.Interaction.Toolkit;
5 using UnityEngine.Events;
6
7 public class MoveWithHand : MonoBehaviour
8 {
9     private XRGrabInteractable grabInteractable;
10    private Collider objectCollider;
11    public AudioSource audioSource;
12
13    private void Start()
14    {
15        // Get the reference to the AudioSource component
16        grabInteractable = GetComponent<XRGrabInteractable>();
17        objectCollider = GetComponent<Collider>();
18
19        grabInteractable.selectEntered.AddListener(new UnityAction<
20        SelectEnterEventArgs>(OnGrabbed));
21        grabInteractable.selectExited.AddListener(new UnityAction<
22        SelectExitEventArgs>(OnReleased));
23    }
24
25    private void OnGrabbed(SelectEnterEventArgs args)
26    {
27        // Play the sound effect
28        audioSource.Play();
29        // Desactiva collider de la pieza al agarrarla
30        objectCollider.enabled = false;
31    }
32
33    private void OnReleased(SelectExitEventArgs args)
34    {
35        // Volver a activar collider de la pieza al soltarla
36        objectCollider.enabled = true;
37    }
38 }

```

Listing 8: *MoveWithHand*

```

1 using System.Collections.Generic;
2 using UnityEngine;
3 using TMPro;
4
5 public class PuzzlePieceSpawner : MonoBehaviour // Va a repartir las piezas cuando
6 {
7     // se cambie la categoria (Solo se ejecuta una vez)

```

```

7 // PuzzlePieceSpawner
8 public GameObject piecePrefab;
9 public int numberOfPieces;
10 public GameObject spawnZone1;
11 public GameObject spawnZone2;
12 public GameObject spawnZone3;
13 public float heightY;
14 public List<Vector3> spawnedPositionsZone1 = new List<Vector3>(); // Es una
    lista vac a al principio
15 public List<Vector3> spawnedPositionsZone2 = new List<Vector3>();
16 public List<Vector3> spawnedPositionsZone3 = new List<Vector3>();
17 public float minDistance; // Distancia m nima entre piezas generadas
18
19 // WordManager
20 private List<string> deportesWords = new List<string>() { "FUTBOL", "TENIS", "
    HOCKEY", "NATACION", "BOXEO", "PINGPONG" };
21 private List<string> frutasWords = new List<string>() { "MANZANA", "BANANA", "
    NARANJA", "MANDARINA", "SANDIA", "FRUTILLA" };
22 private List<string> verdurasWords = new List<string>() { "ZANAHORIA", "
    ESPINACA", "PEPINO", "TOMATE", "BERENJENA", "LECHUGA" };
23 private List<string> animalesWords = new List<string>() { "ELEFANTE", "CABALLO"
    , "GALLINA", "CONEJO", "PERRO", "CAMELLO" };
24
25 public TextMeshProUGUI wordText;
26 public TextMeshProUGUI themeText;
27 public TextMeshProUGUI answerText; // Est escondida en Game Scene
28 public TextMeshProUGUI numberOfRevealedLettersText; // Lo puede elegir el
    usuario en UI
29 public TextMeshProUGUI errorText;
30 public TextMeshProUGUI roundText;
31
32 public void Start()
33 {
34     OnCategoryChanged();
35     errorText.text = "0";
36     roundText.text = "1";
37 }
38
39 private void OnCategoryChanged()
40 {
41     string selectedCategory = PlayerPrefs.GetString("Category"); // Obtiene el
    texto de la categor a seleccionada
42     List<string> selectedWords = GetWordsByCategory(selectedCategory); //
    Obtiene la lista de palabras de la categor a seleccionada
43     Debug.Log(selectedCategory);
44     Debug.Log(selectedWords.Count); // Muestra en consola la cantidad de
    palabras de la categor a seleccionada
45     string selectedWord = selectedWords[0]; // Obtiene la primera palabra de la
    lista de palabras de la categor a seleccionada
46     int numberOfRevealedLetters = PlayerPrefs.GetInt("Dificultad");
47     string underscoreString = GenerateUnderscoreString(selectedWord,
    numberOfRevealedLetters); // Genera una cadena de guiones bajos de la misma
    longitud que la palabra aleatoria
48     wordText.text = underscoreString; // Asigna la cadena de guiones bajos al
    componente TextMeshProUGUI de wordText en Game Screen
49     themeText.text = selectedCategory; // Asigna la categor a seleccionada al
    componente TextMeshProUGUI de themeText en Game Screen

```

```

50     SpawnPieces(selectedWord); // Genera las piezas del puzzle
51 }
52
53
54 private List<string> GetWordsByCategory(string category) // Toma como
55 // par metro una categor a y devuelve su lista de palabras
56 {
57     switch (category)
58     {
59         case "Deportes":
60             return deportesWords;
61         case "Frutas":
62             return frutasWords;
63         case "Verduras":
64             return verdurasWords;
65         case "Animales":
66             Debug.Log("Animales");
67             return animalesWords;
68         default:
69             return new List<string>();
70     }
71 }
72
73 private string GenerateUnderscoreString(string word, int
74 numberOfRevealedLetters)
75 // Toma como par metros una palabra y la cantidad de letras reveladas
76 // Devuelve una cadena de guiones bajos de la misma longitud
77 {
78     string underscoreString = ""; // Cadena vac a para concatenar guiones
79     // bajos con letras reveladas
80     int wordLength = word.Length;
81     List<int> revealedLettersIndexes = new List<int>(); // Lista de ndices de
82     // letras reveladas
83
84     for (int j = 0; j < numberOfRevealedLetters; j++)
85     {
86         int randomIndex = Random.Range(0, wordLength); // Genera un ndice
87         // aleatorio entre 0 y el tama o de la palabra
88         if (!revealedLettersIndexes.Contains(randomIndex)) // Si la lista de
89         // ndices de letras reveladas no contiene el ndice aleatorio, lo agrega
90         {
91             revealedLettersIndexes.Add(randomIndex);
92         }
93         else // Si la lista de ndices de letras reveladas ya contiene el
94         // ndice aleatorio, genera otro ndice aleatorio
95         {
96             j--; // Decrementa el contador para que se vuelva a ejecutar el
97             // ciclo
98         }
99     }
100
101     for (int i = 0; i < wordLength; i++)
102     {
103         if (!revealedLettersIndexes.Contains(i)) // Si la lista de ndices de
104         // letras reveladas no contiene el ndice actual, agrega un gui n bajo
105         {
106             underscoreString += "_";
107         }
108     }
109     return underscoreString + word;
110 }

```

```

98         }
99         else
100         {
101             underscoreString += word[i]; // Si la lista de ndices de letras
reveladas contiene el ndice actual, agrega la letra correspondiente
102         }
103
104         if (i < wordLength - 1) // Si no es la ltima letra, agrega un espacio
105         {
106             underscoreString += " ";
107         }
108     }
109
110     return underscoreString;
111 }
112
113 private List<GameObject> spawnedPieces = new List<GameObject>(); // Lista de
piezas generadas
114 private int spawnZoneCounter = 1;
115
116 public void SpawnPieces(string word) // Genera las piezas del puzzle
117 {
118     answerText.text = word; // Es la respuesta correcta, est escondida en
Game Scene
119
120     foreach (GameObject piece in spawnedPieces) // Destruye las piezas
existentes
121     {
122         if (piece != null)
123             Destroy(piece);
124     }
125     spawnedPieces.Clear(); // Limpia la lista de piezas generadas
126
127     // Genera nuevas piezas
128     int wordIndex = 0; // indice para recorrer las letras de la palabra
129     for (int i = 0; i < numberOfPieces; i++) // Genera el n mero de piezas
especificado en el inspector
130     {
131         GameObject newPiece = Instantiate(piecePrefab); // Crea una pieza copia
de la pieza prefab
132         // GetRandomSpawnPosition devuelve una lista de vectores: el primer
elemento es la posici n y el segundo es la rotaci n
133         newPiece.transform.position = GetRandomSpawnPosition()[0]; // Se le
asigna una posici n aleatoria dentro de una de las 3 zonas de spawn
134         newPiece.transform.rotation = Quaternion.Euler(GetRandomSpawnPosition()
[1]); // Se le asigna una rotacion aleatoria dentro de una de las 3 zonas de
spawn
135
136         // Si quedan letras en la palabra, asignar la siguiente letra
137         if (wordIndex < word.Length)
138         {
139             newPiece.GetComponentInChildren<TMPPro.TextMeshPro>().text = word[
wordIndex].ToString();
140             wordIndex++;
141         }
142     }
143     else
144     {

```



```

144         // Si no quedan letras en la palabra, generar una letra aleatoria
145         string randomLetter = ((char)Random.Range(65, 91)).ToString();
146         newPiece.GetComponentInChildren<TMPPro.TextMeshPro>().text =
randomLetter;
147     }
148
149     // Asignar el tag "PieceClone" a la pieza generada
150     string cloneTag = "PieceClone";
151     newPiece.tag = cloneTag;
152
153     spawnedPieces.Add(newPiece);
154 }
155 }
156
157 private List<Vector3> GetRandomSpawnPosition() // Genera una posici n
aleatoria dentro de una de las 3 zonas de spawn
158 {
159     Bounds bounds; // Bounds = Boundaries of a collider
160     List<Vector3> listaVectores = new List<Vector3>(); // Lista de vectores que
va a devolver la funci n
161
162     switch (spawnZoneCounter) // Este contador va a ir aumentando de 1 a 3
163     {
164         case 1:
165             bounds = spawnZone1.GetComponent<BoxCollider>().bounds; // Obtiene
los 1 mites del box collider de la zona 1
166             listaVectores = GetValidSpawnPosition(bounds, spawnedPositionsZone1
, spawnZoneCounter); // Devuelve una posici n dentro de los 1 mites de la
zona 1
167             break;
168         case 2:
169             bounds = spawnZone2.GetComponent<BoxCollider>().bounds; // Obtiene
los 1 mites del box collider de la zona 2
170             listaVectores = GetValidSpawnPosition(bounds, spawnedPositionsZone2
, spawnZoneCounter); // Devuelve una posici n dentro de los 1 mites de la
zona 2
171             break;
172         case 3:
173             bounds = spawnZone3.GetComponent<BoxCollider>().bounds; // Obtiene
los 1 mites del box collider de la zona 3
174             listaVectores = GetValidSpawnPosition(bounds, spawnedPositionsZone3
, spawnZoneCounter); // Devuelve una posici n dentro de los 1 mites de la
zona 3
175             break;
176         default:
177             bounds = spawnZone1.GetComponent<BoxCollider>().bounds;
178             listaVectores = GetValidSpawnPosition(bounds,
spawnedPositionsZone1, spawnZoneCounter);
179             break;
180     }
181
182     return listaVectores;
183 }
184
185 private List<Vector3> GetValidSpawnPosition(Bounds bounds, List<Vector3>
spawnedPositions, int index)
186 // Toma como par metros los 1 mites de una zona de spawn y la lista de

```

```

187     posiciones ya generadas en esa zona
188     {
189         do
190         {
191             float x = 0.1f;
192             float z = 0.1f;
193             Vector3 rotationVector = new Vector3(0, 0, 0);
194             List<Vector3> listVectors = new List<Vector3>();
195             if (index == 1)
196             {
197                 z = 0.73f;
198                 x = Random.Range(bounds.min.x, bounds.max.x);
199                 rotationVector = new Vector3(0, -180, 0);
200             }
201             else if (index == 2)
202             {
203                 z = -2.77f;
204                 x = Random.Range(bounds.min.x, bounds.max.x);
205                 rotationVector = new Vector3(0, -270, 0);
206             }
207             else if (index == 3)
208             {
209                 x = -4.79f;
210                 z = Random.Range(bounds.min.z, bounds.max.z);
211                 rotationVector = new Vector3(0, -90, 0);
212             }
213             Vector3 spawnPosition = new Vector3(x, heightY, z); // Crea un vector
214             con la posici n generada
215
216             // Verificar si la nueva posicion esta lo suficientemente lejos de las
217             posiciones ya generadas
218             bool isValidPosition = true;
219
220             foreach (Vector3 position in spawnedPositions)
221             {
222                 // Eval a si la distancia entre la posici n nueva y una posici n
223                 ya generada es menor a la distancia m nima
224                 if (Vector3.Distance(spawnPosition, position) < minDistance)
225                 {
226                     isValidPosition = false; // Si la distancia es menor a la
227                     distancia m nima, la posici n no es v lida
228                     break; // Sale del ciclo for
229                 }
230             }
231
232             if (isValidPosition) // Si la posici n es v lida, la devuelve y
233             termina el ciclo do-while
234             {
235                 RaycastHit hit; // Raycast = Rayo que sale de un punto en una
236                 direcci n
237                 if (Physics.Raycast(new Vector3(x, bounds.max.y + 0.1f, z), -
238                 Vector3.up, out hit, Mathf.Infinity))
239                 // Si el rayo que sale de la posici n generada hacia abajo choca
240                 con algo, la posici n es v lida
241                 // Esto es porque la posici n generada est en el aire, y
242                 queremos que est sobre el suelo
243                 {

```

```

234         spawnPosition = hit.point + (Vector3.up * heightY); // La
posici n generada se ajusta para que est  sobre el suelo
235     }
236
237     spawnedPositions.Add(spawnPosition); // Se agrega a la lista de
spawn positions
238     IncrementSpawnZoneCounter(); // Se incrementa el contador de zonas
de spawn
239     listVectors.Add(spawnPosition);
240     listVectors.Add(rotationVector);
241     return listVectors;
242 }
243
244     } while (true); // Este ciclo se va a repetir hasta que se encuentre una
posici n v lida
245 }
246
247 private void IncrementSpawnZoneCounter() // Incrementa el contador de zonas de
spawn
248 {
249     spawnZoneCounter++;
250     if (spawnZoneCounter > 3) // Si ya lleg  a generar una pieza por cada
spawn zone, vuelve a empezar desde la zona 1
251     {
252         spawnZoneCounter = 1;
253     }
254     // Es decir, se va llenando 1-2-3, 1-2-3, 1-2-3, ...
255 }
256
257 }

```

Listing 9: *PuzzlePieceSpawner*

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class SceneTransitionManager : MonoBehaviour
7 {
8     public FadeScreen fadeScreen;
9     public void GoToScene(int sceneIndex)
10    {
11        StartCoroutine(GoToSceneRoutine(sceneIndex));
12    }
13    IEnumerator GoToSceneRoutine(int sceneIndex)
14    {
15        fadeScreen.FadeOut();
16        yield return new WaitForSeconds(fadeScreen.fadeDuration);
17
18        // Carga la escena nueva
19        SceneManager.LoadScene(sceneIndex);
20    }
21    public void GoToSceneAsyncScene()
22    {
23        int sceneIndex = PlayerPrefs.GetInt("SceneNumber");
24        StartCoroutine(GoToSceneAsyncRoutine(sceneIndex));

```

```

25     }
26     public void GoToSceneAsync(int sceneIndex)
27     {
28         StartCoroutine(GoToSceneAsyncRoutine(sceneIndex));
29     }
30
31     IEnumerator GoToSceneAsyncRoutine(int sceneIndex)
32     {
33         fadeScreen.FadeOut();
34         AsyncOperation operation = SceneManager.LoadSceneAsync(sceneIndex);
35         operation.allowSceneActivation = false;
36
37         float timer = 0;
38         while (timer <= fadeScreen.fadeDuration && !operation.isDone)
39         {
40             timer += Time.deltaTime;
41             yield return null;
42         }
43
44         operation.allowSceneActivation = true;
45     }
46 }

```

Listing 10: *SceneTransitionManager*

```

1  using UnityEngine;
2  using UnityEngine.UI;
3
4  public class SetBackgroundVolume : MonoBehaviour
5  {
6      public Slider volumeSlider;
7      public AudioSource audioSource1;
8      public AudioSource audioSource2;
9
10     private float initialVolume1;
11     private float initialVolume2;
12
13     private void Start()
14     {
15         // Guarda los valores iniciales del volumen
16         initialVolume1 = audioSource1.volume;
17         initialVolume2 = audioSource2.volume;
18     }
19
20     public void OnVolumeChanged()
21     {
22         float sliderValue = volumeSlider.value;
23
24         // Calcula el volumen nuevo para cada AudioSource basado en el valor del
25         // slider.
26         float newVolume1 = initialVolume1 * sliderValue;
27         float newVolume2 = initialVolume2 * sliderValue;
28
29         // Setea el volumen nuevo para cada AudioSource.
30         audioSource1.volume = newVolume1;
31         audioSource2.volume = newVolume2;
32     }
33 }

```

32 }

Listing 11: *SetBackgroundVolume*

```

1 using UnityEngine;
2 using UnityEngine.UI;
3 using TMPro;
4
5 public class SetNumberOfRevealedLetters : MonoBehaviour
6 {
7     public Slider slider;
8     public TextMeshProUGUI numberOfRevealedCharacters;
9
10    public void Start()
11    {
12        slider.value = int.Parse(numberOfRevealedCharacters.text); // Setea el
13        valor inicial del slider
14    }
15
16    public void OnNumberChanged()
17    {
18        // Esta funci n se invoca cuando el valor del slider cambia
19        int value = (int)slider.value;
20        numberOfRevealedCharacters.text = value.ToString();
21    }
22 }
```

Listing 12: *SetNumberOfRevealedLetters*

```

1 using UnityEngine;
2 using UnityEngine.UI;
3
4 public class SetSoundVolume : MonoBehaviour
5 {
6     public Slider volumeSlider;
7     public AudioSource audioSource1;
8     public AudioSource audioSource2;
9     public AudioSource audioSource3;
10    public AudioSource audioSource4;
11    public AudioSource audioSource5;
12
13    private float initialVolume1;
14    private float initialVolume2;
15    private float initialVolume3;
16    private float initialVolume4;
17    private float initialVolume5;
18
19    private void Start()
20    {
21        initialVolume1 = audioSource1.volume;
22        initialVolume2 = audioSource2.volume;
23        initialVolume3 = audioSource3.volume;
24        initialVolume4 = audioSource4.volume;
25        initialVolume5 = audioSource5.volume;
26    }
27
28    public void OnVolumeChanged()
```

```

29     {
30         float sliderValue = volumeSlider.value;
31
32         float newVolume1 = initialVolume1 * sliderValue;
33         float newVolume2 = initialVolume2 * sliderValue;
34         float newVolume3 = initialVolume3 * sliderValue;
35         float newVolume4 = initialVolume4 * sliderValue;
36         float newVolume5 = initialVolume5 * sliderValue;
37
38         audioSource1.volume = newVolume1;
39         audioSource2.volume = newVolume2;
40         audioSource3.volume = newVolume3;
41         audioSource4.volume = newVolume4;
42         audioSource5.volume = newVolume5;
43     }
44 }

```

Listing 13: *SetSoundVolume*

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using TMPro;
6  using System.Linq;
7
8  public class ToggleGroupSelectionManager : MonoBehaviour
9  {
10     public ToggleGroup toggleGroup1; // Toggle para elegir el tema de palabras
11     public ToggleGroup toggleGroup2; // Toggle para elegir la escena
12     public TextMeshProUGUI dificultad;
13
14     public void SelectCategory()
15     {
16         // Selecci n del tema de palabras
17         Toggle toggle1 = toggleGroup1.ActiveToggles().FirstOrDefault();
18         string selectedCategory = toggle1.GetComponentInChildren<Text>().text;
19         PlayerPrefs.SetString("Category", selectedCategory);
20
21         // Selecci n del nivel de dificultad
22         if (dificultad.text == "1")
23         {
24             PlayerPrefs.SetInt("Dificultad", 3);
25         }
26         else if (dificultad.text == "2")
27         {
28             PlayerPrefs.SetInt("Dificultad", 2);
29         }
30         else if (dificultad.text == "3")
31         {
32             PlayerPrefs.SetInt("Dificultad", 1);
33         }
34         else
35         {
36             PlayerPrefs.SetInt("Dificultad", 0);
37         }
38     }

```

```

39     // Selecci n de la escena
40     Toggle toggle2 = toggleGroup2.ActiveToggles().FirstOrDefault();
41     string selectedScene = toggle2.GetComponentInChildren<Text>().text;
42     if (selectedScene == "Soleado")
43     {
44         PlayerPrefs.SetInt("SceneNumber", 1);
45     }
46     else if (selectedScene == "Nevado")
47     {
48         PlayerPrefs.SetInt("SceneNumber", 2);
49     }
50     else if (selectedScene == "Lluvioso")
51     {
52         PlayerPrefs.SetInt("SceneNumber", 3);
53     }
54     else
55     {
56         PlayerPrefs.SetInt("SceneNumber", 0);
57     }
58 }
59 }

```

Listing 14: *ToggleGroupSelectionManager*

```

1  using UnityEngine;
2  using UnityEngine.UI;
3  using System.Linq;
4
5  public class ToggleSystem : MonoBehaviour
6  {
7      ToggleGroup toggleGroup;
8      void Start()
9      {
10         toggleGroup = GetComponent<ToggleGroup>();
11         // Por default el primer toggle est  seleccionado
12         toggleGroup.SetAllTogglesOff(); // Asegurar que no haya ning n toggle
seleccionado
13         toggleGroup.transform.GetChild(0).GetComponent<Toggle>().isOn = true;
14     }
15
16     public void OnThemeChanged()
17     {
18         Toggle toggle = toggleGroup.ActiveToggles().FirstOrDefault(); // Obtiene el
toggle activado
19         Debug.Log(toggle.name + " _ " + toggle.GetComponentInChildren<Text>().text)
; // Imprime el nombre del toggle y el texto del label
20     }
21 }

```

Listing 15: *ToggleSystem*

9.2. Fase de testeo

CONSENTIMIENTO INFORMADO

"Aplicación de Realidad Virtual para el entrenamiento físico-funcional de los pacientes con la Enfermedad de Parkinson"

El que suscribe,, DNI N°., mayor de edad, manifiesta que ha tomado conocimiento que en el Departamento de Bioingeniería del Instituto Tecnológico de Buenos Aires (ITBA) se va a desarrollar un Proyecto Final de Grado denominado: **"Aplicación de Realidad Virtual para el entrenamiento físico-funcional de los pacientes con la Enfermedad de Parkinson"**, que es llevado a cabo por la estudiante de la carrera de Bioingeniería del ITBA Jungmin Koh y asistido por sus tutores, el Dr. Fabián Eduardo Cremaschi y los Ing. Santiago Bestani y Santiago Racca, y al respecto expresa:

- 1.- Que dicho Proyecto consiste en adquirir y estudiar los siguientes parámetros relacionados con la marcha, en personas sanas: la cantidad de pasos, la cantidad de giros y la distancia recorrida. El voluntario debe completar los cuestionarios ofrecidos al finalizar la prueba, para tomar datos sobre el cybersickness y la usabilidad del sistema. En el caso de verificarse la efectividad del proyecto, el estudio aportará importante información para los pacientes en tratamiento.
- 2.- Que el Proyecto tiene por finalidad desarrollar una aplicación de realidad virtual para rehabilitación físico-funcional de pacientes con Parkinson. De este modo se podrá entrenar al paciente para caminar y resolver problemas cognitivos simultáneamente, lo que contribuirá favorablemente al avance del tratamiento y al mejoramiento del trastorno de la marcha parkinsoniana.
- 3.- Que para la realización del Proyecto es necesario contar con la colaboración de personas que estén dispuestas a participar en él como voluntarios y que se les pedirá no más de 60 minutos de su tiempo.
- 4.- Que el que suscribe debe usar los dispositivos de realidad virtual, que consiste en un casco con visores y dos controladores de manos. Este estudio es no invasivo, no genera dolor o molestias y no posee efectos secundarios a largo plazo. Existe la rara posibilidad de presentar, de manera temporal, mareos, náuseas, desorientación y fatiga visual relacionados con el efecto de cybersickness. Para ello se lo conducirá a un laboratorio de captura de movimientos del ITBA, donde se le solicitará colocarse el equipo de realidad virtual y tendrá unos minutos de aclimatación para disminuir el riesgo de los malestares.
- 5.- Que el voluntario puede desistir de seguir participando en el Proyecto, cuando así lo desee.
- 6.- Que la participación del voluntario es enteramente gratuita, no pudiendo reclamar ningún tipo de remuneración, viático ni lucro cesante.
- 7.- Que toda la información que le ha sido dada al voluntario, la ha recibido contando con la posibilidad de expresar sus dudas e inquietudes, las que han sido atendidas por el informante a través del diálogo directo y personal que ha podido tener con ella.
- 8.- Que la decisión del voluntario de participar en el Proyecto es producto de su exclusiva voluntad, la que ha manifestado libremente y con pleno conocimiento de los hechos por la adecuada información y comprensión que de ellos ha podido tener, dejando expresa constancia de no haber sido inducido de manera alguna por el informante a expresar el consentimiento.
- 9.- Que manifiesta haber comprendido todo lo expuesto y que todas sus dudas fueron perfectamente aclaradas, y que participa voluntariamente del presente Proyecto.
- 10.- Que por todo lo precedentemente expuesto y afirmado, el voluntario presta su consentimiento para participar en el presente Proyecto.

Figura 9.1: Consentimiento informado para la prueba

System Usability Scale

© Digital Equipment Corporation, 1986.

	Strongly disagree							Strongly agree
1. I think that I would like to use this system frequently	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
	1	2	3	4	5			
2. I found the system unnecessarily complex	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
	1	2	3	4	5			
3. I thought the system was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
	1	2	3	4	5			
4. I think that I would need the support of a technical person to be able to use this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
	1	2	3	4	5			
5. I found the various functions in this system were well integrated	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
	1	2	3	4	5			
6. I thought there was too much inconsistency in this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
	1	2	3	4	5			
7. I would imagine that most people would learn to use this system very quickly	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
	1	2	3	4	5			
8. I found the system very cumbersome to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
	1	2	3	4	5			
9. I felt very confident using the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
	1	2	3	4	5			
10. I needed to learn a lot of things before I could get going with this system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			
	1	2	3	4	5			

Figura 9.2: Sistema de Escala de Usabilidad original en inglés [116]

Sistema de Escala de Usabilidad (SUS)

Este cuestionario es una herramienta estandarizada para medir de forma universal la percepción de la usabilidad de un sistema. Por favor, del 1 al 5, marque con un círculo la respuesta que mejor corresponda a su experiencia con el simulador de realidad virtual **BrainWalk VR**.

1. Me gustaría usar este sistema con frecuencia:

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

2. El sistema me resultó innecesariamente complejo.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

3. El sistema fue fácil de usar.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

4. Sentí la necesidad de ayuda de un técnico para usar este sistema.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

5. Varias funciones en este sistema me resultaron bien integradas.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

Figura 9.3: Sistema de Escala de Usabilidad traducido en español (página 1)

6. El sistema me resultó demasiado inconsistente o incoherente.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

7. Creo que la mayoría de personas aprendería a usar este sistema muy rápidamente.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

8. El sistema me resultó muy incómodo para usar.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

9. Me sentí muy seguro/a usando el sistema.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

10. Tuve que aprender varias cosas antes de poder usar este sistema.

1	2	3	4	5
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

Figura 9.4: Sistema de Escala de Usabilidad traducido en español (página 2)

CyberSickness in Virtual Reality Questionnaire (CSQ-VR)

A brief tool for evaluating the Virtual Reality Induced Symptoms and Effects (VRISE)

Please, from 1 to 7, **circle** the response that better corresponds to the presence and intensity of the symptom.

Nausea A: Do you experience nausea (e.g., stomach pain, acid reflux, or tension to vomit)?

1	2	3	4	5	6	7
Absent Feeling	Very Mild Feeling	Mild Feeling	Moderate Feeling	Intense Feeling	Very Intense Feeling	Extreme Feeling

Please write below any additional comments relevant to the question above:

Nausea B: Do you experience dizziness (e.g., light-headedness or spinning feeling)?

1	2	3	4	5	6	7
Absent Feeling	Very Mild Feeling	Mild Feeling	Moderate Feeling	Intense Feeling	Very Intense Feeling	Extreme Feeling

Please write below any additional comments relevant to the question above:

Vestibular A: Do you experience disorientation (e.g., spatial confusion or vertigo)?

1	2	3	4	5	6	7
Absent Feeling	Very Mild Feeling	Mild Feeling	Moderate Feeling	Intense Feeling	Very Intense Feeling	Extreme Feeling

Please write below any additional comments relevant to the question above:

Vestibular B: Do you experience postural instability (i.e., imbalance)?

1	2	3	4	5	6	7
Absent Feeling	Very Mild Feeling	Mild Feeling	Moderate Feeling	Intense Feeling	Very Intense Feeling	Extreme Feeling

Please write below any additional comments relevant to the question above:

Figura 9.5: Cuestionario de Cíbermareo en Realidad Virtual original en inglés (página 1) [119]

Oculomotor A: Do you experience a visually induced fatigue (e.g., feeling of tiredness or sleepiness)?

1	2	3	4	5	6	7
Absent Feeling	Very Mild Feeling	Mild Feeling	Moderate Feeling	Intense Feeling	Very Intense Feeling	Extreme Feeling

Please write below any additional comments relevant to the question above:

Oculomotor B: Do you experience a visually induced discomfort (e.g., eyestrain, blurred vision, or headache)?

1	2	3	4	5	6	7
Absent Feeling	Very Mild Feeling	Mild Feeling	Moderate Feeling	Intense Feeling	Very Intense Feeling	Extreme Feeling

Please write below any additional comments relevant to the question above:

Category	Symptom	Symptom Intensity	Category Score
Nausea	Nausea (Nausea A)		
	Dizziness (Nausea B)		
Vestibular	Disorientation (Vestibular A)		
	Imbalance (Vestibular B)		
Oculomotor	Fatigue (Oculomotor A)		
	Discomfort (Oculomotor B)		
CSQ-VR Score =			

Symptom Intensity = the score provided by the responder.

Category Score = Score A + Score B

CSQ-VR Score = Nausea score + Vestibular score + Oculomotor score

CyberSickness in Virtual Reality Questionnaire (CSQ-VR) derives from the VR Neuroscience Questionnaire (VRNQ).

Both were developed by Panagiotis Kourtesis.

Figura 9.6: Cuestionario de Cibermareo en Realidad Virtual original en inglés (página 2) [119]

Cuestionario de Cibermareo en Realidad Virtual (CSQ-VR)

Esta encuesta se trata de una herramienta breve para evaluar los Síntomas y Efectos Inducidos por la Realidad Virtual (VRISE). Por favor, del 1 al 7, marque con un círculo la respuesta que mejor corresponda a la presencia e intensidad del síntoma.

Náuseas A: ¿Experimenta náuseas (por ejemplo, dolor de estómago, reflujo ácido o sensación de querer vomitar)?

1	2	3	4	5	6	7
Ausente	Muy Leve	Leve	Moderado	Intenso	Muy Intenso	Extremo

Comentario adicional:

Náuseas B: ¿Experimenta mareos (por ejemplo, sensación de mareo o sensación de giro)?

1	2	3	4	5	6	7
Ausente	Muy Leve	Leve	Moderado	Intenso	Muy Intenso	Extremo

Comentario adicional:

Vestibular A: ¿Experimenta desorientación (por ejemplo, confusión espacial o vértigo)?

1	2	3	4	5	6	7
Ausente	Muy Leve	Leve	Moderado	Intenso	Muy Intenso	Extremo

Comentario adicional:

Vestibular B: ¿Experimenta inestabilidad postural (es decir, desequilibrio)?

1	2	3	4	5	6	7
Ausente	Muy Leve	Leve	Moderado	Intenso	Muy Intenso	Extremo

Comentario adicional:

Figura 9.7: Cuestionario de Cibermareo en Realidad Virtual traducido en español (página 1)

Oculomotor A: ¿Experimenta fatiga por la vista (por ejemplo, sensación de cansancio o somnolencia)?

1	2	3	4	5	6	7
Ausente	Muy Leve	Leve	Moderado	Intenso	Muy Intenso	Extremo

Comentario adicional:

Oculomotor B: ¿Experimenta molestias por la vista (por ejemplo, cansancio ocular, visión borrosa o dolor de cabeza)?

1	2	3	4	5	6	7
Ausente	Muy Leve	Leve	Moderado	Intenso	Muy Intenso	Extremo

Comentario adicional:

Resultado:

Categoría	Síntoma	Intensidad del síntoma	Puntuación por categoría
Náuseas	Náuseas (A)		
	Mareo (B)		
Vestibular	Desorientación (A)		
	Desequilibrio (B)		
Oculomotor	Fatiga (A)		
	Molestias (B)		
		Puntuación CSQ - VR	

Intensidad del síntoma = Puntuación marcada por el encuestado

Puntuación por categoría = Puntuación A + Puntuación B

Puntuación CSQ-VR = Puntuación Náuseas + Puntuación Vestibular + Puntuación Oculomotor

Figura 9.8: Cuestionario de Cibermareo en Realidad Virtual traducido en español (página 2)

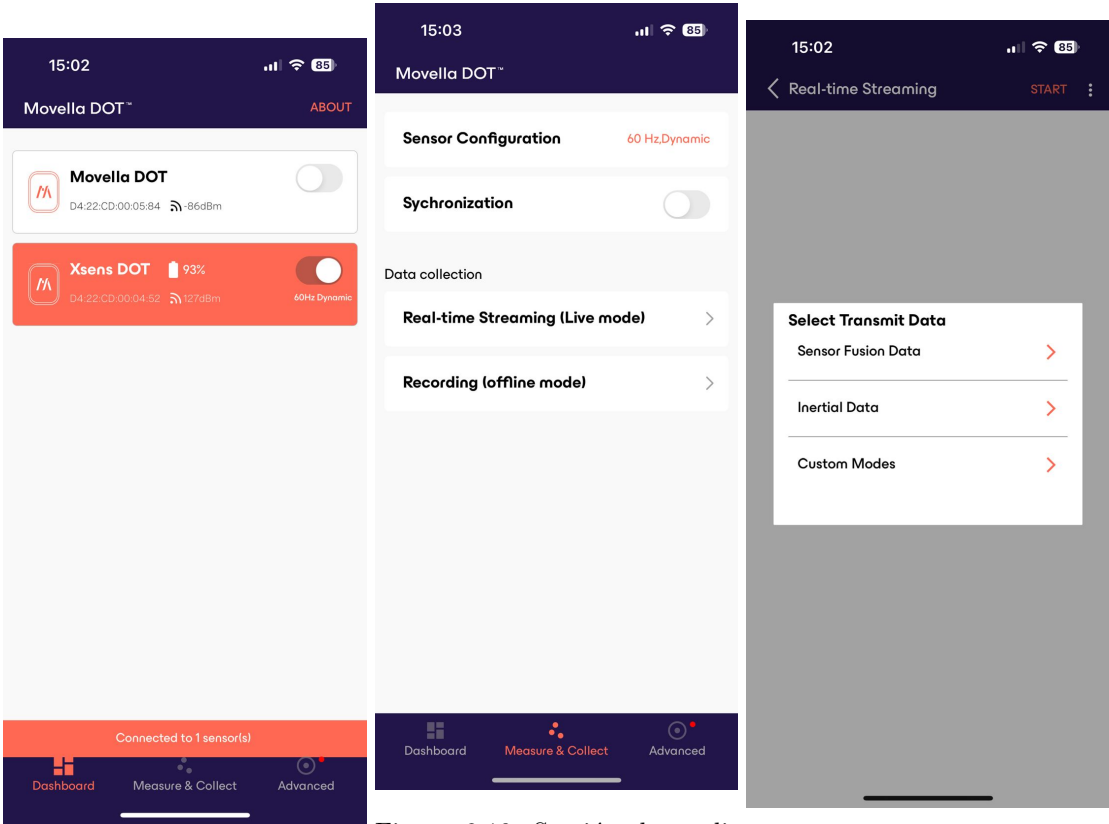


Figura 9.9: Conexión del sensor a la aplicación por Bluetooth

Figura 9.10: Sección de mediciones y recolección de datos. En el proyecto se eligió trabajar con *Real-time Streaming*

Figura 9.11: Modos de transmisión de datos en *Real-time Streaming*

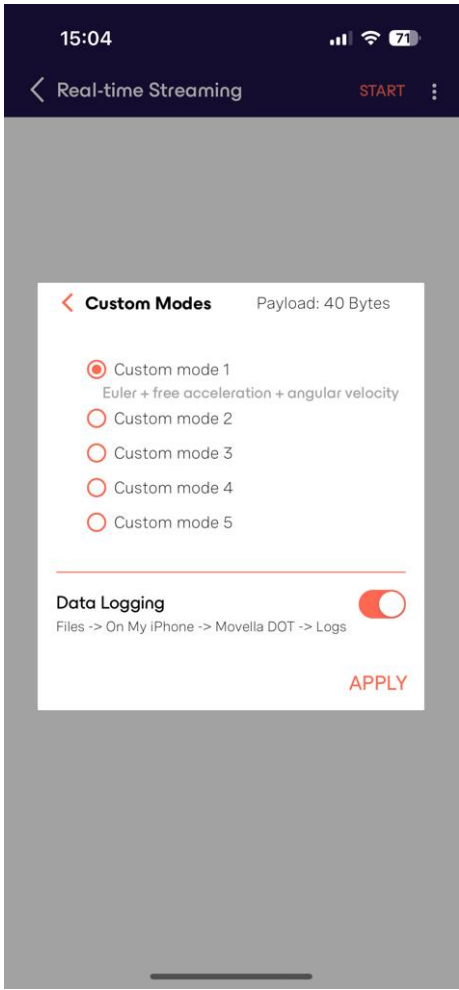


Figura 9.12: Diferentes modos customizados para las mediciones

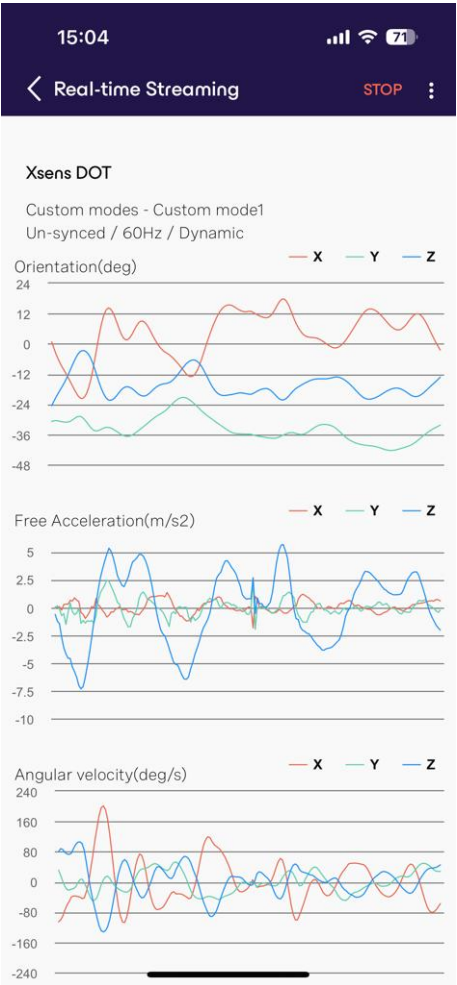


Figura 9.13: Visualización de datos transmitidos en tiempo real. Se miden la orientación en ángulos de Euler, la aceleración libre, y la velocidad angular (desde arriba hacia abajo)