

Ambiente Virtual Para Rehabilitación de Motricidad Fina en Miembro Superior Utilizando Sensor de Movimiento *LeapMotion*[®]



**Emma Alejandra Chaves Guerrero
Nelson Andrés Ortiz Bolaños**

Director:
PhD. Carlos Felipe Rengifo Rodas

Universidad del Cauca
**Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Ingeniería Electrónica y Telecomunicaciones**
Popayán, Noviembre de 2014

Ambiente Virtual Para Rehabilitación de Motricidad Fina en Miembro Superior Utilizando Sensor de Movimiento *LeapMotion*[®]



Trabajo de Grado presentado como requisito para optar al título de Ingeniero en Electrónica y Telecomunicaciones

Emma Alejandra Chaves Guerrero
Nelson Andrés Ortiz Bolaños

Director:
PhD. Carlos Felipe Rengifo Rodas

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Ingeniería Electrónica y Telecomunicaciones
Popayán, Noviembre de 2014

TABLA DE CONTENIDO

CAPÍTULO 1.....	1
Conceptos preliminares	1
1.1 Conceptos sobre habilidades motrices y motricidad fina	1
1.2 Conceptos de Rehabilitación y Rehabilitación virtual.....	3
1.3. Realidad virtual.	4
1.4. Sistemas sensores movimiento.....	9
1.5. Trabajos relacionados a la aplicación de realidad virtual utilizando sensores de movimiento en rehabilitación	10
CAPÍTULO 2.....	17
Herramientas Hardware y Software.....	17
2.1. Dispositivo LeapMotion.....	17
2.2. LeapMotion SDK.....	18
2.3. API LeapMotion	20
2.4. Selección de herramientas Software	23
CAPÍTULO 3.....	31
TRAYECTORIAS	31
3.1. Trazado de trayectorias en 2D	32
3.2. Trayectorias en 3D del ambiente virtual aplicando el método de curvas de Bézier...36	
3.3. Modelo matemático para el renderizado de la trayectoria 3D del ambiente virtual ...37	
3.4. Calculo de la orientación del Toroide	39
3.5. Modelo matemático para el renderizado del toroide del ambiente virtual	41
CAPÍTULO 4.....	43
DESCRIPCIÓN DE LA INTERFAZ GRÁFICA Y AMBIENTE VIRTUAL DEL PROTOTIPO DESARROLLADO.....	43
1.1. Descripción de la Interfaz gráfica de Usuario.....	44
4.2. Descripción ambiente virtual.....	58
CAPÍTULO 5.....	67
PRUEBAS DE EVALUACIÓN Y ANALISIS DEL AMBIENTE VIRTUAL	67
5.1. Consideraciones para la prueba de evaluación del ambiente virtual	67
5.2. Pruebas de evaluación del ambiente virtual	68

5.3. Registro de los datos de la prueba del ambiente virtual	70
5.4. Análisis de los resultados	71
CONCLUSIONES	75
TRABAJOS FUTUROS Y RECOMENDACIONES	77
BIBLIOGRAFÍA.....	79
ANEXO A	1
INSTALACIÓN Y CONFIGURACIÓN DEL DISPOSITIVO LEAPMOTION	1
ANEXO B	5
PROCEDIMIENTO DE INSTALACIÓN DE LA APLICACIÓN CON TODOS SUS REQUERIMIENTOS	5

Índice de figuras

Figura 1.1. Imagen de visita virtual, Abu Simbel, 1250 A.C. Tomado de Enciclopedia Microsoft Encarta 2009.	7
Figura 1.2. “CAVE”. Tomado de http://www.antycipsimulation.com	8
Figura 1.3. Configuración del Controlador LeapMotion. Tomado de [18]	10
Figura 1.4. Interacción del usuario con una aplicación de realidad virtual a través del dispositivo sensor LeapMotion. Fuente propia	10
Figura 1.5. Entorno sistema de rehabilitación Kinect. Tomado de [20].....	12
Figura 1.6. Sistema de interacción basado CyWee Z. Tomado de [22].....	13
Figura 1.7. Configuración, consistente en el LeapMotion Controller, y un robot industrial (Kuka Robot KR 125/3): (a) Vista frontal y (b) Vista esquemática de un sistema de coordenadas. Tomado de [23].....	14
Figura 1.8. Sistema para análisis de LeapMotion basado en captura óptica de QualisysTrack. Tomado de [24]	15
Figura 2.1. Sistema de coordenadas del dispositivo LeapMotion Controller. Tomado de www.leapmotion.com	18
Figura 2.2. Arquitectura del sistema LeapMotion	20
Figura 3.1. Trayecto de B(s), curva de Bézier de primer orden. Tomado de: Animated linear Bézier curve (Philip Tregoning)	34
Figura 3.2. Trayecto de B(s), curva de Bézier de segundo orden. Tomado de: Animated quadratic Bézier curve (Philip Tregoning)	35
Figura 3.3. Trayecto de B(s), curva de Bézier de cuarto orden. Tomado de: Animated fourth order Bézier curve (Philip Tregoning).....	35

Figura 3.4. Representación espacial del vértice de control P usando coordenadas esféricas. Fuente propia	37
Figura 3.5. Posicionamiento del cilindro en el segmento generado entre S_0 y S_1 . Fuente propia	38
Figura 3.6. Representación de la trayectoria en 3D en el ambiente virtual. Fuente propia	39
Figura 3.7. Orientación del Toroide. Fuente propia	40
Figura 3.8. Posición y Dirección del Toroide. Fuente propia	41
Figura 3.9. Desplazamientos de los ángulos β y α para la construcción del toroide. Tomado de Wikipedia	42
Figura 3.10. Ubicación de cada rectángulo en el volumen del toroide. Tomado de Wikipedia	42
Figura 4.1. Metodología XP. a) Ciclo de desarrollo o iteración y b) Ciclo de vida metodología XP. Tomado de [29]	44
Figura 4.2. Diagrama de casos de uso del sistema. Fuente propia	46
Figura 4.3. Diagrama de Clases del sistema. Fuente propia	47
Figura 4.4. Grafo del árbol de JAVA 3D. Fuente propia	48
Figura 4.5. Interfaz gráfica de usuario. Fuente propia	49
Figura 4.6. Interfaz registro pacientes. Fuente propia	50
Figura 4.7. Interfaz registro paciente ingresando número de documento. Fuente propia	50
Figura 4.8. Registro de pacientes almacenado previamente en la base de datos. Fuente propia	51
Figura 4.9. Interfaz evaluación paciente. Fuente propia	52
Figura 4.10. Interfaz observaciones. Fuente propia	53
Figura 4.11. Interfaz consultar datos pacientes. Fuente propia	54
Figura 4.12. Interfaz consulta datos pacientes probando el campo de texto "Nombre". Fuente propia	54
Figura 4.13. Interfaz consulta datos pacientes probando el campo de texto "Número Documento". Fuente propia	55
Figura 4.14. Interfaz gráfica de resultados evaluación pacientes. Fuente propia	56
Figura 4.15. Interfaz opción imprimir evaluación. Fuente propia	56
Figura 4.16. Resultado desempeño paciente. Fuente propia	57
Figura 4.17. Resultado Desempeño Pacientes sigma (Desviación Estándar). Fuente propia	57
Figura 4.18. Opción DEMO en el ambiente virtual. Fuente propia	58
Figura 4.19. Opción "practicas" en el ambiente virtual. Fuente propia	59
Figura 4.20. "Iniciar Sesión" para la práctica de ejercicios en el ambiente virtual. Fuente propia	59
Figura 4.21. Ventana de diálogo advirtiendo que el usuario no se encuentra registrado. Fuente propia	60
Figura 4.22. Inicio de sesión para un usuario registrado en el sistema. Fuente propia	60

Figura 4.23. Trayectoria generada aleatoriamente, en el nivel de complejidad 1, teniendo activo cursor toroide. Fuente propia	61
Figura 4.24. Trayectoria generada aleatoriamente, la esfera verde indica que se encuentra recorriendo la curva. Fuente propia.....	62
Figura 4.25. Trayectoria generada aleatoriamente, la esfera roja indica que se encuentra por fuera de la curva. Fuente propia	62
Figura 4.26. Trayectoria generada aleatoriamente con nivel de complejidad 1. Fuente propia	63
Figura 4.27. Trayectoria generada aleatoriamente con nivel de complejidad 3. Fuente propia	63
Figura 4.28. Trayectoria generada aleatoriamente con nivel de complejidad 8. Fuente propia	64
Figura 4.29. Trayectoria generada aleatoriamente con nivel de complejidad 8, toroide. Fuente propia.....	64
Figura 4.30. Trayectoria generada aleatoriamente con nivel de complejidad 10. Fuente propia	65
Figura 4.31. "Cerrar Sesión" para la práctica de ejercicios en el ambiente virtual. Fuente propia	65
Figura 5.1. Trayectorias de prueba. a) Nivel 1 de complejidad, b) Nivel 5 de complejidad, c) Nivel 7 de complejidad. Fuente propia	69
Figura 5.2. Desviación estándar al cabo de diez días para el nivel 1	72
Figura 5.3. Desviación estándar al cabo de diez días para el nivel 5	73
Figura 5.4. Desviación estándar al cabo de diez días para el nivel 7	73

Índice de tablas

<i>Tabla 1.1. Aspectos considerados en Realidad Virtual.</i>	8
<i>Tabla 2.1. Unidades de medida de las cantidades físicas.....</i>	18
<i>Tabla 2.2. Descripción clases básicas del API LeapMotion.</i>	23
<i>Tabla 2.3 Metodologías de desarrollo de software tradicionales.....</i>	29
<i>Tabla 5.1. Tiempo (mm:ss.ms) y desviación estándar (m.m) nivel 1</i>	70
<i>Tabla 5.2. Tiempo (mm:ss.ms) y desviación estándar (m.m) nivel 5</i>	71
<i>Tabla 5.3. Tiempo (mm:ss.ms) y desviación estándar (m.m) nivel 7</i>	71
<i>Tabla 5.4. Comparación de la Media cinco primeros días y cinco restantes para nivel 1</i>	74
<i>Tabla 5.5. Comparación de la Media cinco primeros días y cinco restantes para nivel 5</i>	74
<i>Tabla 5.6. Comparación de la Media cinco primeros días y cinco restantes para nivel 7.....</i>	74

CAPÍTULO 1

CONCEPTOS PRELIMINARES

La motricidad fina involucra movimientos pequeños de la mano y muñeca de gran precisión que son utilizados particularmente en acciones donde se usan de manera simultánea la coordinación del ojo, mano (oculo-manual) y dedos, dichas acciones pueden ser realizadas en la vida cotidiana como lo son: pintar, enhebrar, escribir, etc.

Las dificultades de motricidad en el ser humano son originadas por diferentes causas: van desde accidentes cerebrovasculares, trastornos de la motricidad en el desarrollo, sobreesfuerzo muscular de la pinza digital, mano o muñeca hasta degeneraciones neurológicas [1]. Dichas dificultades necesitan de tratamiento médico especializado con el propósito de lograr la rehabilitación del paciente.

Los avances de la tecnología y la computación aplicada en el área de la salud, permiten desarrollar entornos de realidad virtual con el fin de relacionarlos al tratamiento de rehabilitación motriz, integrándose con los procedimientos convencionales. Por medio de un entorno de realidad virtual se pueden realizar ejercicios que ayuden al proceso de rehabilitación motriz fina. Dicho entorno puede ser práctico, agradable, intuitivo y alentador. También permite que los ejercicios del tratamiento se puedan adaptar y personalizar para satisfacer las necesidades de cada paciente.

En este primer capítulo del presente trabajo de grado se exponen conceptos que comprende la rehabilitación virtual. Se presentan los conceptos generales en cuanto a rehabilitación motriz fina y realidad virtual. Además, en el desarrollo de la aplicación descrita en este trabajo, se emplea el sensor de movimiento *LeapMotion*[®] para la interacción del paciente con el ambiente virtual, también se muestran los conceptos propios de los sistemas basados en sensores de movimiento. Por último, se mencionan algunas aplicaciones relacionadas con los propósitos de este trabajo.

1.1 Conceptos sobre habilidades motrices y motricidad fina

“Por habilidad motriz entendemos la competencia (grado de éxito o de consecución de las finalidades propuestas) de un sujeto frente a un objetivo dado, aceptando que, para la consecución de este objetivo, la generación de

respuestas motoras, el movimiento desempeña un papel primordial e insustituible” [2]. De lo anterior, se puede definir a las habilidades motrices como las actividades que permiten a los individuos efectuar movimientos finos (son movimientos que requieren de coordinación de lo que se ve y se toca, involucrando el ojo, mano, muñeca y dedos) y gruesos (se refiere al desplazamiento, al cambio de posición, reacciones posturales y de equilibrio). Cabe resaltar que son aquellas habilidades ejecutadas en la vida cotidiana, como lo es: escribir una carta, hacer la cama, un sastre que confecciona una pieza de ropa hasta un partido de baloncesto, son ejemplos claros de habilidades motrices.

Además, la motricidad es la función que considera actividades de tres sistemas que regulan el movimiento y son: Sistema Piramidal, Sistema Extrapiramidal y el Sistema Cerebeloso. El primero se encarga de los movimientos voluntarios del cuerpo humano, el segundo de la motricidad automática y el último regula la armonía del equilibrio interno de los movimientos, el trabajo en conjunto de estos tres sistemas conforman la habilidad motriz [3].

Como se había mencionado, la habilidad motriz está comprendida por dos grandes categorías: la motricidad gruesa y la motricidad fina. Por lo tanto, la motricidad humana lleva una secuencia de movimientos amplios y precisos que los realizará a lo largo de toda su vida. La motricidad gruesa implica los movimientos en brazos, piernas, tronco, esta es la base primordial para la motricidad fina ya que de ella se derivan las destrezas que se adquieren para realizar actividades cotidianas como lo son: escribir, enhebra, etc.

“Los trastornos de la motricidad involucran anormalidades leves o graves del tono muscular, postura, movimiento y adquisición de habilidades motrices” [1]. Los trastornos más comunes de la motricidad fina conllevan a problemas con la escritura o trazos previos a ésta, estos son más comunes en los primeros años de vida. Una manera de detectar estos problemas es la disgrafía, que es la mala escritura, esta dificultad se presenta sin necesidad de que exista un trastorno neurológico o intelectual que lo evidencie [4]. Existen ejercicios que favorecen el desempeño de la motricidad fina y por ende a la calidad de la escritura como: las actividades para desarrollar la destreza de las manos y dedos, actividades para desarrollar la coordinación óculo manual. Sobre estos aspectos se centra la atención de la asistencia médica, que tiene como propósito la rehabilitación de motricidad fina en los pacientes.

1.2 Conceptos de Rehabilitación y Rehabilitación virtual

La definición de rehabilitación según la RAE¹ es: *“Med. Conjunto de métodos que tiene por finalidad la recuperación de una actividad o función perdida o disminuida por traumatismo o enfermedad.”* [5].

“La Rehabilitación es un proceso global y continuo de duración limitada y con objetivos definidos, encaminados a promover y lograr niveles óptimos de independencia física y las habilidades funcionales de las personas con discapacidades, como así también su ajuste psicológico, social, vocacional y económico que le permitan llevar de forma libre e independiente su propia vida.” [6]. La rehabilitación es un proceso complejo que resulta de la aplicación integrada de muchos procedimientos para lograr que el individuo recupere su estado funcional óptimo, tanto en el hogar como en la comunidad, en la medida que lo permita la utilización apropiada de todas sus capacidades residuales.

La OMS (Organización Mundial de la Salud) define: *“la rehabilitación y la habilitación son procesos destinados a permitir que las personas con discapacidad alcancen y mantengan un nivel óptimo de desempeño físico, sensorial, intelectual, psicológico y/o social. La rehabilitación abarca un amplio abanico de actividades, como atención médica de rehabilitación, fisioterapia, psicoterapia, terapia del lenguaje, terapia ocupacional y servicios de apoyo”* [7].

La rehabilitación como la expone la Dra. Ana Grasielle Dionísio en [8] cumple con el objetivo de desarrollar la máxima funcionalidad y calidad de vida, la rehabilitación física (o fisioterapia), el diseño de programas, la selección, la planificación y el uso de modalidades técnicas, educativas y terapéuticas específicas basadas en movimiento, las terapias de manipulación y los recursos naturales y físicos, elegido a partir del análisis y evaluación de movimiento y postura.

La rehabilitación motriz busca recuperar las carencias motrices que generalmente son caracterizadas por alteraciones en las funciones del cuerpo que interfieren en el movimiento y/o locomoción de las personas [9]. Por esta razón, las funciones podrían afectar el desenvolvimiento de las personas en la ejecución de sus labores diarias, es decir la rehabilitación se considera un proceso que resulta del estudio completo de muchos procedimientos con el propósito de lograr que las personas recobren su estado funcional óptimo. Hay varios tipos de rehabilitación y son: rehabilitación física, rehabilitación cognitiva, musicoterapia, práctica mental y rehabilitación virtual. Si bien estos tipos de procedimientos terapéuticos usan técnicas y elementos diferentes para cumplir con los objetivos de rehabilitación, estos tienen el propósito de estimularlas

¹ Real Academia Española

funciones mentales que incluyen: la atención, percepción, comprensión, aprendizaje, memoria y raciocinio entre otros. Las anteriores, son funciones de la cognición que ayudan a la realización de secuencias motoras complejas necesarias para el desarrollo de acciones como: cocer un pantalón, pintar un cuadro y tocar un instrumento musical. De acuerdo con la temática desarrollada en la presente monografía, es necesario abordar temas como la rehabilitación virtual y todos los aspectos relacionados con ella.

La rehabilitación virtual brinda al paciente la sensación de estar en una situación real con el propósito de interactuar en el entorno. Además, *“El realismo de un entorno virtual está determinado por: resolución y fidelidad de la imagen, reproducción de las propiedades de los objetos y de los escenarios virtuales, las respuestas de los objetos deben reaccionar igual que un objeto real e interactividad”* [10].

De acuerdo con lo anterior, la incursión de nuevas tecnologías ha influido de manera positiva en diversos campos de la salud, significativamente se pueden observar grandes aportes en el tema de la rehabilitación cognitiva y motora donde la aplicación de entornos virtuales genera apoyo a los procedimientos convencionales de tratamiento haciéndolos más eficaces y eficientes [11].

“La aplicación de realidad virtual en terapias para rehabilitación motora, es una de las más innovadoras para el área de fisioterapia terapéutica, desde el punto de vista de volver el proceso más placentero, por medio de juegos e interfaces no convencionales con un computador” [12].

La respuesta sensorial en un entorno virtual debe permitirle al usuario percibir tanto la firmeza o flexibilidad de un objeto virtual, así como la inclusión de sonidos en la escena virtual. La realidad virtual es una experiencia inmersiva e interactiva a través de una herramienta computacional que sucede en tiempo real y que permite visualizar respuestas tridimensionales a ciertos movimientos.

1.3. Realidad virtual.

Es importante sintetizar de las múltiples definiciones existentes de realidad virtual, una que involucre, relacione y exprese con claridad distintos aspectos, tal como: *“Realidad Virtual (RV) es una interface de usuario avanzada para aplicaciones computacionales, que permiten al usuario navegar e interactuar, en tiempo real, con un ambiente tridimensional generado por computador, usando dispositivos multisensoriales”* [13].

El inicio de la realidad virtual se dio en el entrenamiento y simulación de vuelo de los pilotos del ejército, pero las aplicaciones de RV en entornos médicos tuvo gran crecimiento, ya que anteriormente las prácticas y pruebas de cirugías,

se realizaban directamente en pacientes reales. Por lo que la realidad virtual se torna como un método estándar que permite manipular el ciclo de acción y percepción del comportamiento normal subyacente, con lo cual se logra que los experimentos se centren en la comparación y análisis de los resultados obtenidos en el mundo virtual para aplicarlos en el mundo real.

La realidad virtual en la medicina se introdujo para educación y entrenamiento de médicos, la exploración de sus habilidades a través de simuladores quirúrgicos prueba la enseñanza cognitiva y las habilidades manuales de cirujanos y residentes. Actualmente, los simuladores quirúrgicos son utilizados en oftalmología, urología, ortopedia, cirugía neurológica, cirugía cardíaca, entre otras [14].

Se ha demostrado con éxito el beneficio potencial de integrar tecnología de realidad virtual en múltiples aspectos de la medicina y la rehabilitación, donde presentan oportunidades que facilitan las prácticas de recuperación de las capacidades perdidas, incluyendo la motricidad fina. Además, los entornos con realidad virtual permiten mantener el control y la medida sobre la entrega experimental de ciertos estímulos.

La realidad virtual también ofrece la capacidad de individualizar las necesidades de tratamiento, mientras que proporciona una mayor estandarización de los protocolos de evaluación y capacitación. De hecho, la evidencia preliminar indica que la realidad virtual ofrece un medio único en el que la terapia puede ser proporcionada dentro de un contexto funcional, útil y motivador, donde se puede clasificar y documentar los resultados fácilmente [15].

Al sintetizar en un concepto lo que es la Realidad Virtual no se contemplan todos los aspectos que esta incluye. Por tanto, se toman características generales de los sistemas de realidad virtual con el fin de estandarizar el concepto, estas características se muestran en la siguiente sección.

1.3.1. Aspectos generales de la Realidad Virtual

En términos de adaptación del usuario a la realidad virtual, se definen los conceptos de *participación*, *interacción* e *inmersión*, de tal manera que se caracterice la relación entre el entorno y el usuario, para generar respuestas cognitivas que sean propias de situaciones reales, pero que son simuladas por computador.

Participación

La concepción de participación se centra en el efecto de motivar al usuario con un entorno agradable, por tanto “participación es la capacidad de estimular la motivación para que el usuario participe con el ambiente virtual” [14], además la

reproducción de realidad en un ambiente que recopile elementos atrayentes, estimulan la curiosidad del usuario y lo llevan a simular sus actividades habituales en el escenario virtual. En consecuencia, la participación puede ser pasiva y/o activa, ya que la realidad virtual tiene el potencial para trabajar los dos tipos:

- Pasiva: cuando solo se observa el desarrollo del ambiente de forma regida sobre rutas trazadas.
- Activa: cuando se realiza una actividad conjunta, que implique variaciones en componentes del ambiente.

Interacción

La realidad virtual es interactiva porque el usuario no se limita a ser un observador pasivo del ambiente, al contrario, toma control para realizar acciones que son detectadas por el sistema y se manifiestan en el ambiente virtual, modificándolo en el curso de la simulación. El ambiente es caracterizado por la forma como reacciona a las acciones y modificaciones del usuario, lo cual es un factor que pone a prueba las condiciones intuitivas de manipulación y navegación que se aplican en el mundo real.

Los niveles de interacción se definen de tres maneras: *pasiva*, *exploratoria* o *interactiva*. A continuación se muestra una descripción de cada una.

1. La realidad virtual pasiva: el usuario es capaz de observar, escuchar o hasta sentir el ambiente en movimiento en torno de sí mismo, lo que causa una impresión del usuario de que se está moviendo en lugar del ambiente. Entre tanto la navegación por el mundo sintético es controlada automáticamente por el computador, de manera que el usuario es apenas un espectador.
2. La realidad virtual exploratoria: permite que el mismo usuario explore y se mueva por el ambiente virtual, sin embargo, el usuario no es capaz de interactuar de otra forma con el ambiente, por ejemplo no podría manipular objetos que estén dentro de la escena. Una muestra de aplicación de realidad virtual exploratoria son las visitas virtuales 3D, como las presentadas en la Enciclopedia Microsoft Encarta 2009, en la que se realiza la exploración mediante el uso del mouse, ver Figura 1.1
3. La realidad interactiva: es más realista, pues posibilita que además de navegar, el usuario también modifique el ambiente a través de sus acciones. En este caso se puede observar la capacidad reactiva del ambiente y sus entidades. Por ejemplo en un carro virtual, si el usuario presiona el botón de encender la radio del carro, el computador debe generar sonidos simulando una emisora.



Figura 1.1. Imagen de visita virtual, Abu Simbel, 1250 A.C. Tomado de Enciclopedia Microsoft Encarta 2009.

Inmersión

La idea de la inmersión es conseguir que el usuario tenga la sensación de encontrarse presente en el mundo virtual, *“formar parte del ambiente”*. Se pueden obtener varios niveles de inmersión dependiendo de los dispositivos hardware que se utilicen, por lo que el concepto de inmersión clasifica la realidad virtual en: Inmersiva y No-Inmersiva. Además, hay una forma intermedia denominada realidad virtual semi-inmersiva, la cual utiliza recursos que promueven cierto grado de inmersión al usuario.

La realidad virtual no-inmersiva hace uso solamente de equipos convencionales como: monitor, teclado, mouse y software, que permiten la implantación de movimientos a los usuarios en el entorno, dando la impresión de virtualidad en el sistema.

La realidad virtual inmersiva busca la inmersión total del usuario en el ambiente y es obtenida a través de equipos no convencionales como casco de visualización o cavernas virtuales *“CAVE”* (ver Figura 1.2); y los dispositivos ligados a otros sentidos que también son importantes para sentir la inmersión, como dispositivos de retorno de fuerza o sonidos etc.

Este nivel de inmersión del usuario permite que las habilidades y conocimientos intuitivos puedan ser utilizados en la manipulación de objetos virtuales, obteniendo como resultado la sensación de presencia y estar actuando dentro

del ambiente virtual, apuntando, golpeando, manipulando o ejecutando otras acciones sobre los objetos virtuales, en tiempo real.



Figura 1.2. "CAVE". Tomado de <http://www.antycipsimulation.com>

1.3.2. Consideraciones en el diseño de una aplicación con ambientes virtuales

El desarrollo de ambientes virtuales requiere tener en consideración formas atractivas y aspectos de respuesta a las expectativas del usuario. Claudio Kirner [16], indica que un ambiente virtual debería cumplir con algunos de los requisitos enunciados en la Tabla 1.1.

Sintético	El ambiente virtual debe ser generado en tiempo real por el sistema computacional, a diferencia de lo que no ocurre con algunos sistemas multimedia, en los cuales las presentaciones son grabadas previamente a su ejecución.
Tridimensional	El ambiente debe hacer uso de recursos que estimulen diferentes sentidos humanos, como visión, audición, tacto, percepción del espacio, profundidad etc.
Inmersión	El ambiente debe proporcionar al usuario una sensación de cierto grado de inmersión.
Interactivo	El sistema debe permitir la detección de estímulos (solicitudes) enviados por el usuario, instantáneamente, promover alteraciones y/o acciones en escenas y objetos del mundo sintético.
Presencia	Se caracteriza por tener un sentido subjetivo de que el usuario está físicamente en el ambiente, y algunas veces participando activamente de este ambiente.

Tabla 1.1. Aspectos considerados en Realidad Virtual.

1.4. Sistemas sensores movimiento

Como se ha planteado, la inmersión es parte fundamental en los ambientes virtuales, por ende, vale la pena describir algunos conceptos de los dispositivos de interacción basados en sensores de movimiento, que son útiles en sistemas de realidad virtual. Algunos dispositivos pueden ser complejos, pero permiten transmitir información entre el usuario y el computador dando sensación de presencia o alto grado de realidad en la interacción con el ambiente.

Los sensores de seguimiento de movimiento pueden medir e informar la posición y orientación de un objeto a medida que este cambia. Un ejemplo sencillo de seguimiento de movimiento es el cursor en una pantalla de computador, donde el movimiento del cursor está en correspondencia con el desplazamiento del ratón. Pero, el seguimiento en 3D requiere la utilización de dispositivos que adapten un espacio volumétrico del mundo real al espacio virtual de la escena. Lograr dichas características no es fácil, es por ello que existen técnicas como el *Motion Capture* que son capaces de seguir el movimiento de la cabeza, brazos, manos y demás partes del usuario. Como identifican en [17], el Motion Capture tiene la capacidad de ofrecer análisis cuantitativo, obteniendo datos fiables y mediciones precisas. Las anteriores, son características a tener en cuenta en este trabajo sobre rehabilitación de motricidad fina y en la evaluación clínica de las habilidades motoras, ya que es interesante tener la posibilidad de capturar y medir cuantitativamente una variedad de movimientos del miembro superior en escenarios de rehabilitación de la vida diaria.

La reducción del tamaño, el peso y la conectividad de los dispositivos actuales, cumplen con requisitos de practicidad para ser implementados en sistemas de rehabilitación, logrando que la evaluación cuantitativa de movimiento pueda ser soportada en escalas de evaluación clínica estándar y proporcionando a los médicos información enriquecida sobre la salud de los pacientes.

Algunos de estos dispositivos se basan en captura de movimientos por cámara, giróscopos, acelerómetros, etc. De estos, el controlador LeapMotion en conjunto con el API (Application Programmer Interface), proporciona la posición en el espacio cartesiano de objetos predefinidos, como los dedos, manos, gestos de manos, puntas de lápices, etc. Las posiciones entregadas están referenciadas al punto central del controlador LeapMotion, el cual se encuentra centrado en la posición del segundo emisor infrarrojo (ver: Figura 1.3-b), además, el controlador consta de tres emisores IR (luz infrarroja) y dos cámaras IR, por lo que éste se puede clasificar en los sistemas de seguimiento ópticos basados en la visión estéreo [18].

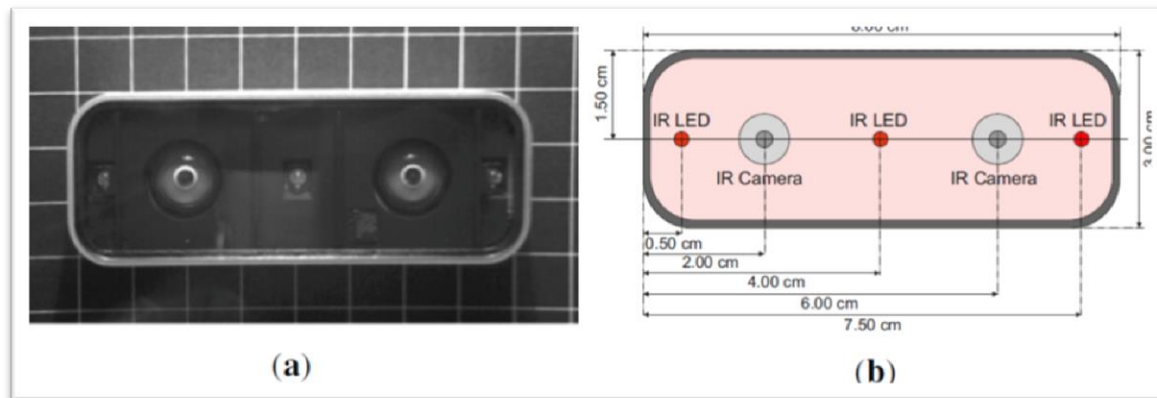


Figura 1.3. Configuración del Controlador LeapMotion. Tomado de [18]

Las características particulares del dispositivo LeapMotion, permiten al usuario realizar movimientos de interacción en ambientes virtuales. Además de tener la retroalimentación visual, se consigue el efecto de desplazamiento en el espacio real (actuación ver: Figura 1.4), concluyendo en una interacción rápida e intuitiva con el computador y el espacio virtual.

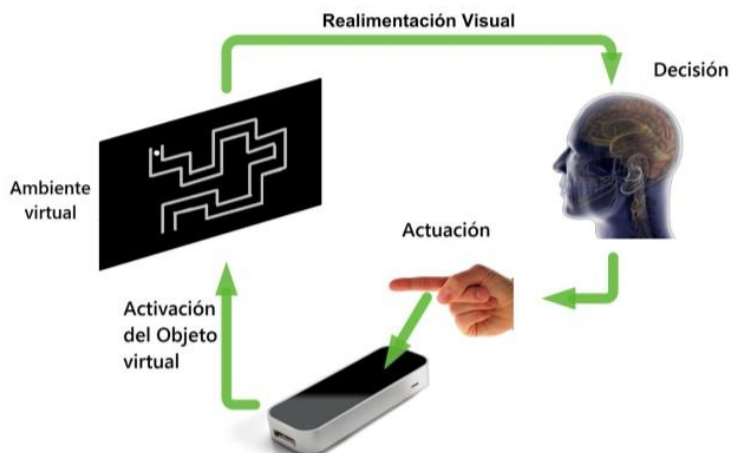


Figura 1.4. Interacción del usuario con una aplicación de realidad virtual a través del dispositivo sensor LeapMotion. Fuente propia

1.5. Trabajos relacionados a la aplicación de realidad virtual utilizando sensores de movimiento en rehabilitación

Las aplicaciones de realidad virtual, requieren de un medio que permita al ser humano interactuar con ellas; por ejemplo las interfaces hápticas y los sensores de movimiento, que son dispositivos desarrollados expresamente para captar los movimientos del cuerpo.

Las interfaces hápticas activas, presentan características generales de sensores y actuadores, donde estas interfaces además de leer la posición y movimientos del usuario, también le transmiten las fuerzas resultantes de su interacción con el entorno. Además, existen tipos de interfaces hápticas multi-dedo basadas en estructuras tipo guantes, exoesqueletos y los manipuladores que se sostiene suspendidos en el techo. *“Los sistemas sostenidos sobre el*

techo o un muro no son sistemas portátiles, mientras que los exoesqueletos y las interfaces que actúan sobre la mano del operador si suelen serlo”, además, “el desempeño de las interfaces hápticas, en sistemas como exoesqueletos donde el usuario debe soportar todo el peso de la estructura sobre su mano o el brazo dificulta la manipulación de objetos y el desarrollo de tareas colaborativas”. Entonces, se pueden tener en cuenta criterios esenciales para la rehabilitación como la portabilidad y usabilidad que otros dispositivos pueden cubrir, de esta manera se puede aumentar el beneficio y comodidad del paciente a costos razonables [19].

Por otra parte, los entornos virtuales pueden ser llevados más allá de la finalidad prevista, y convertir un sistema de juegos en una herramienta tecnológica de apoyo en la rehabilitación. A continuación, se realiza una breve descripción de algunos dispositivos que se han implementado con fines médicos en el área de rehabilitación. Así como se describe en el estudio del Departamento de Ingeniería Electrónica y el centro de desarrollo de Dispositivos Médicos Holísticos de la Universidad Cristiana Chung Yuan de Taiwan [20]. Se evalúa la posibilidad de rehabilitación de pacientes con parálisis cerebral usando un sistema basado en Kinect. Ya que, tiene como objetivo utilizar la tecnología de procesamiento de imágenes en el diseño de un sistema para ayudar a motivar a las personas con discapacidad motora en el aumento del número de ejercicios, la mejora del dominio motor y la calidad de vida.

El sistema consta de un Kinect de Microsoft, que se conecta a un ordenador portátil de IBM T61 con Microsoft Windows 7, en el que se instala el software de rehabilitación desarrollado. Además, cuenta con un módulo de audio que se utilizó para entregar retroalimentación de audio y la pantalla como interacción visual. Por otra parte, los participantes actuaban al frente del sistema a aproximadamente 3 metros delante del módulo de Kinect (Figura 1.5), utilizando las señales en la pantalla para completar el programa prescrito. El sistema detecta los movimientos de los participantes durante las actividades de rehabilitación y determina si corresponden a los requisitos de los terapeutas.



Figura 1.5. Entorno sistema de rehabilitación Kinect. Tomado de [20]

En Colombia se ha desarrollado la implementación del dispositivo Kinect de Microsoft para PC con el fin de generar trayectorias para un robot PUMA [21] simulado a partir de la captura del movimiento de la mano, de la que se obtienen las coordenadas cartesianas de la trayectoria realizada por el movimiento de la mano del usuario. Además, la herramienta desarrollada en este caso para el dispositivo está compuesta por una interfaz de usuario que permite visualizar la ejecución del robot PUMA 762 simulado y aumentar la velocidad en ejecución de la trayectoria realizada.

Sin embargo, se presentaron problemas de baja resolución en el sensor Kinect, ya que es necesario establecer un entorno propicio de iluminación en el área de trabajo, dado que *“la luz directa a los lentes de las cámaras del Kinect dificulta el reconocimiento del usuario y no es posible la captura del movimiento humano”*.

Por otro lado, la calidad geométrica de los datos de profundidad obtenidos por el sensor de Kinect indican que el error aleatorio de medición de la profundidad aumenta con el aumento de la distancia al sensor y va desde unos pocos milímetros hasta aproximadamente 4 cm en el rango máximo del sensor. La precisión de los datos también se encuentra influenciada por la baja resolución de las mediciones de profundidad. El dispositivo Kinect de Microsoft para PC, no es un dispositivo de alta precisión (por debajo de 0.2 mm).

El *Grupo de Investigación Neurológica R.E.A.L* en el Centro de Investigación de Fisioterapia de la Universidad de Otago en Dunedin, Nueva Zelanda, desarrolló

un auto-soporte bilateral para las extremidades superiores con intervención en rehabilitación basada en un dispositivo de juego (**¡Error! La autoreferencia al marcador no es válida.**), que ejercita los movimiento en personas con accidente cerebrovascular crónico [22]. La intervención consiste en jugar en un computador con el módulo de movimiento CyWee Z² (grupo CyWee Ltd; Taipei, Taiwan), que se incorporó a un manillar a medida. Con este dispositivo, tres participantes jugaron en tres PCs con uno o dos terapeutas supervisores, con la misma cantidad de interacción terapéutica proporcionada durante un tratamiento de control. Además se utilizaron un conjunto de juegos de complejidad gradual como objetivo estratégico del juego (Bejeweled y BalloonPopping), otros con distintos niveles de impacto (Mosquito Swat, Music Catch, y ReBounce), y unos con mayor rapidez de interacción.



Figura 1.6. Sistema de interacción basado CyWee Z. Tomado de [22]

Este dispositivo presenta una desventaja, debido a que algunos participantes mencionaron dolor en el hombro durante y después de la intervención. Este dolor no les impidió participar, pero debe tenerse en cuenta el peso y lo

²El módulo de movimiento CyWee puede seguir el movimiento con posición absoluta, el sentido de profundidad en el espacio de 3 dimensiones y se satisfacen para aplicación basada en la localización, de juego y de reconocimiento de gestos.

concerniente al controlador cuando se utiliza con más frecuencia, para una mayor duración, o sin supervisión.

Por otra parte, el Departamento de Computación y Ciencia, de la Universidad Técnica de Dortmund, Alemania, realizó un análisis de la precisión y robustez del controlador LeapMotion [23]. Dado que, en el desarrollo de sistemas con interfaces de usuario basadas en captura de gestos de la mano humana, la precisión alcanzable de la medición del movimiento es un factor muy relevante. Los sistemas de interacción se ven afectados por el llamado temblor, que se define como un movimiento involuntario y rítmico de los músculos, y dependiendo de la edad humana, “*la amplitud del temblor varía entre 0.4 mm \pm 0.2 mm para los individuos jóvenes y 1.1 mm \pm 0.6 mm para las personas mayores*”.

En este estudio, se utilizó un sistema de referencia robótico de gran precisión y calidad, con el fin de evaluar la precisión y la repetibilidad del controlador *LeapMotion*. Estos robots industriales presentan una precisión en la gama sub-milimétrica, y sirven como sistema de referencia, ya que supera la precisión del temblor humano “por debajo de 0.2 mm”. El robot industrial utilizado en la prueba es el KUKA KR 125/3, que proporciona una precisión repetible de menos de 0.2 mm. En la Figura 1.7 se muestra la configuración de la prueba que consiste en el controlador *LeapMotion* y el robot industrial.

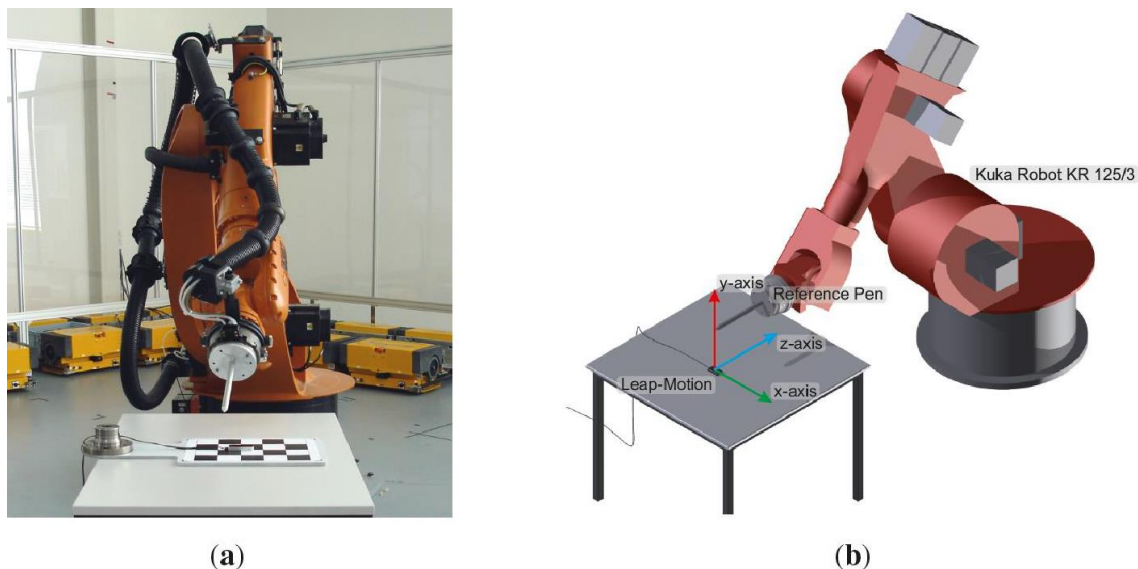


Figura 1.7. Configuración, consistente en el LeapMotion Controller, y un robot industrial (Kuka Robot KR 125/3): (a) Vista frontal y (b) Vista esquemática de un sistema de coordenadas. Tomado de [23]

con el montaje, se concluyó que la precisión de la medición para configuraciones estáticas de la posición deseada y las posiciones medidas es

de menos de 0.2 mm. La evaluación para situaciones dinámicas, presentó valores con precisión menores de 2.5 mm, con una media de 1.2 mm.

Como análisis complementario al estudio anterior, en la Facultad de Ingeniería Eléctrica de la Universidad de Liubliana en Eslovenia, se realizó una investigación para determinar precisión y fiabilidad, pero en función de la distorsión espacial y la consistencia de la frecuencia de muestreo del controlador *LeapMotion* en condiciones estáticas y dinámicas [24], con el fin de determinar la idoneidad del sensor de seguimiento de mano, gestos y objetos. La evaluación se realizó con la ayuda de un sistema de captura de movimiento óptico de alta velocidad y gran precisión.

El sistema óptico de referencia calibrado, consta de ocho cámaras de alta velocidad Oqus3+ y el software de administrador QualisysTrack de la empresa Sueca Qualisys-Inc, que proporciona un volumen de medición de aproximadamente 1 m³, con una resolución de 1,3 millones de píxeles y una frecuencia de cuadro constante de 500 fotogramas por segundo (Figura 1.8).



Figura 1.8. Sistema para análisis de LeapMotion basado en captura óptica de QualisysTrack. Tomado de [24]

Del análisis se concluye que, para el escenario estático, la desviación estándar muestra ser inferior a 0,5 mm en todo momento, en los mejores casos menos de 0,01 mm. Además, se comprueba la alta precisión (debajo de 0,2 mm) que se informó en [22] [23] para el controlador como un sistema fiable y preciso para el seguimiento de puntos estáticos. Por otro lado, el conjunto de medidas en el escenario dinámico reveló inconsistencias en el rendimiento del controlador en comparación al sistema óptico. Su precisión se evaluó a través de la distorsión de la distancia entre dos puntos en movimiento, con una distancia entre puntos constante.

Por tanto, se puede decir que si bien el controlador *LeapMotion* no tiene el mismo alcance de un sistema óptico de captura de movimiento rápido y de alta precisión en un espacio limitado y con un número limitado de objetos, basándose en los resultados obtenidos y la experiencia general, se demuestra que el controlador en el estado actual podría ser utilizado en sistemas de seguimiento no profesional, ya que las prestaciones son muy buenas en comparación al costo de un sistema óptico de alta precisión.

CAPÍTULO 2

HERRAMIENTAS HARDWARE Y SOFTWARE

En este capítulo se hace una descripción de las características del hardware y software a utilizar para el desarrollo del presente trabajo de grado. La herramienta hardware a usar es el dispositivo electrónico basado en sensores de movimiento *LeapMotion*. *Este dispositivo cuenta con un software que permite la captura de datos del movimiento de los objetos dentro del campo visual.*

Una de las herramientas software que se describe en este capítulo y que sirve para el diseño del prototipo propuesto en el presente trabajo de grado es la implementación de *JAVA* y su entorno de desarrollo *NetBeans*. Además, para administrar, almacenar los datos obtenidos por el dispositivo *LeapMotion* es necesario el uso de un sistema de gestión de bases datos relacional *MySQL*. Por otra parte, se presenta algunas herramientas complementarias empleadas en el diseño del prototipo, como lo es la escogencia de la metodología de desarrollo de Software.

2.1. Dispositivo LeapMotion

El dispositivo *LeapMotion* es un sensor de movimiento de alta precisión, con muy buena tasa de velocidad de localización que detecta, rastrea las manos y los dedos. Además, utiliza un software interno que analiza los objetos presentes en el campo de vista, obteniendo tanto posiciones discretas, gestos y movimiento de las manos. El campo de vista del *LeapMotion* es una pirámide invertida centrada en el dispositivo, el cual tiene un alcance efectivo que se extiende desde aproximadamente 25 hasta 600 milímetros por encima del dispositivo [25].

2.1.1. Sistema de coordenadas del dispositivo

LeapMotion utiliza un sistema de coordenadas cartesiano diestro, donde el origen se centra en la parte superior del dispositivo. La distribución espacial como se muestra en la Figura 2.1, ubica los ejes X y Z en el plano horizontal, donde el eje de las X que corre paralelo al borde largo del dispositivo y el eje Z tiene valores positivos crecientes hacia el usuario. En tanto que el eje Y es vertical, con valores positivos crecientes hacia arriba.

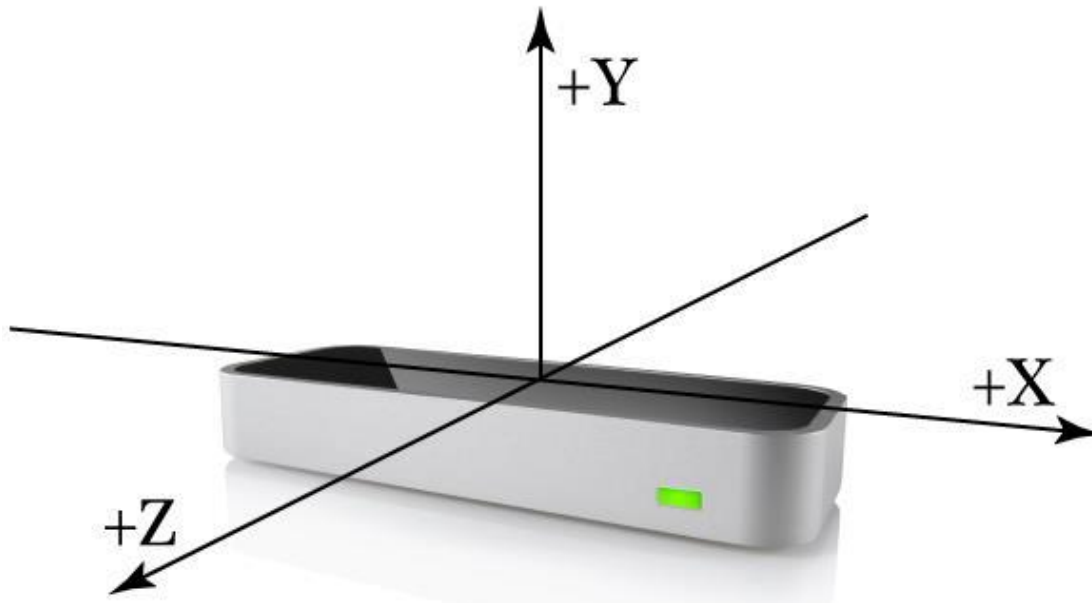


Figura 2.1. Sistema de coordenadas del dispositivo LeapMotion Controller. Tomado de www.leapmotion.com

Por otra parte, el dispositivo tiene en cuenta algunas cantidades físicas que son las que determinan la precisión de los movimientos. En la Tabla 2.1 se muestra dichas cantidades con sus respectivas unidades de medida:

Cantidad Física	Unidad de Medida
Distancia:	Milímetros
Tiempo:	microsegundos (a menos que se indique lo contrario)
Velocidad:	milímetros/segundo
Ángulo de visión:	radianes

Tabla 2.1. Unidades de medida de las cantidades físicas.

2.2. LeapMotion SDK

La biblioteca de *LeapMotion* está escrita en lenguaje C++. También, proporciona *SWIG* que es una herramienta de código abierto utilizada para conectar las bibliotecas escritas en lenguajes como: C #, Java y Python a la base de la biblioteca de C++ de *LeapMotion*. Cabe resaltar, que por cada *SWIG* se usan dos bibliotecas adicionales. Además, para JavaScript y desarrollo de aplicaciones web el *LeapMotion* proporciona un servidor WebSocket y una biblioteca de JavaScript del lado del cliente.

Por otro lado, LeapMotion SDK incluye todos los archivos, bibliotecas, códigos y cabeceras, que son necesarios para desarrollar aplicaciones y plug-ing. Excepto la librería de cliente JavaScript *leap.js*.

2.2.1. Java SDK

Dentro de Leap.jar están contenidas las clases de Java para LeapMotion. Este código carga las librerías libLeapJava.dylib para Mac, LeapJava.dll para Windows, o libLeapJava.so para Linux. Estas bibliotecas contienen el código nativo que traduce los llamados de Java a la base del API de LeapMotion en libLeap.dylib, Leap.dll o libLeap.so dependiendo de la plataforma.

2.2.2. Arquitectura del Sistema

El software del dispositivo LeapMotion se ejecuta como un servicio (en Windows) o demonio (en Mac y Linux), este se comunica con el dispositivo a través del bus USB para acceder al servicio y recibir los datos de seguimiento de movimiento. El SDK de LeapMotion proporciona dos variedades de la API para obtener los datos de movimiento: una interfaz nativa y una interfaz de WebSocket. Estas API permiten crear aplicaciones habilitadas para el dispositivo en varios lenguajes de programación, incluyendo JavaScript que se ejecuta en un entorno de navegador.

El servicio / demonio de salto de movimiento se comunica con las aplicaciones que utilizan TCP en la IP localhost (127.0.0.1) y puertos: 6437, 6438 y 6439 Esta dirección y puertos no pueden ser bloqueados por un firewall u otro programa.

Además, se proporciona una interfaz de aplicación nativa a través de una biblioteca cargada dinámicamente. Esta biblioteca se conecta al servicio y proporciona datos de seguimiento a la aplicación creada. Puede enlazar a la biblioteca por medio de uno de los enlaces de lenguaje previstos en Java, C # y Python.

El servicio de *LeapMotion* (Leap Service, ver Figura 2.2), recibe datos desde el controlador a través del bus USB, este servicio procesa la información y la envía a *Leap-enabledapplications*. Por defecto, el servicio sólo envía los datos de seguimiento a la aplicación en primer plano (*foreground*). Sin embargo, las aplicaciones pueden solicitar que se reciban los datos en segundo plano (*background*), y la petición puede ser negada por el usuario.

La aplicación de LeapMotion se ejecuta por separado del servicio y permite que el usuario pueda configurar la instalación. La aplicación LeapMotion es un applet de panel de control en Windows y una aplicación de barra de menús en Mac OS X.

La aplicación *foreground Leap-enabled* recibe los datos de seguimiento del movimiento del servicio. Una aplicación *Leap-enabled* puede conectarse al servicio LeapMotion utilizando la biblioteca nativa LeapMotion.

Cuando una aplicación *Leap-enabled* pierde el foco del sistema operativo, el servicio LeapMotion detiene el envío de datos a la misma. Las aplicaciones destinadas a trabajar en segundo plano (*background*) pueden solicitar permiso para recibir los datos, incluso cuando está en *background*. Cuando está en *background*, las opciones de configuración están determinadas por la aplicación *foreground*.

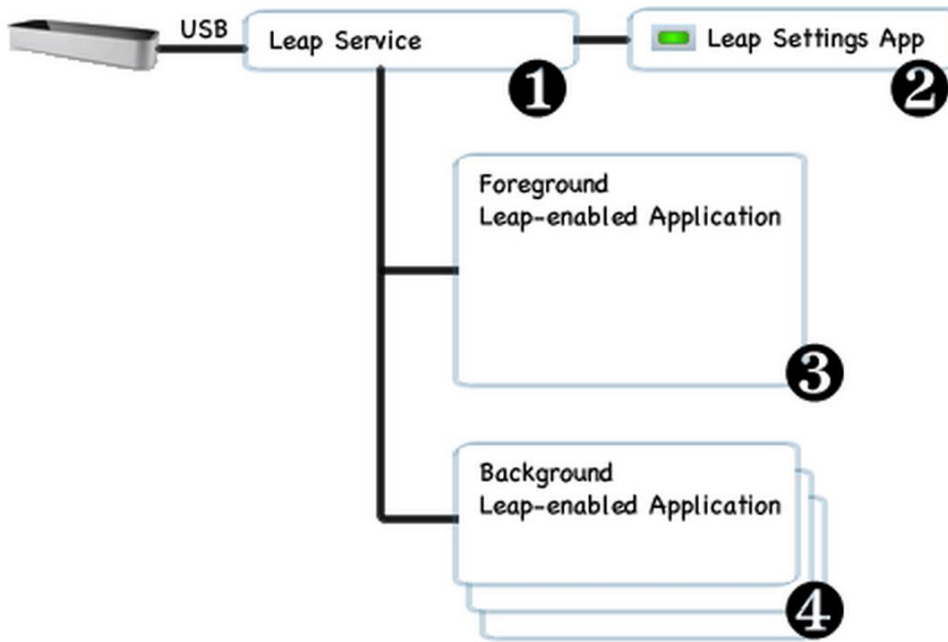


Figura 2.2. Arquitectura del sistema LeapMotion

2.3. API LeapMotion

En el sitio web de los desarrolladores del dispositivo *LeapMotion* [25] se encuentra el *API Reference*, que es una base de información sobre todas las clases que componen la API *LeapMotion*. Además, algunas clases permiten obtener datos de movimiento que el dispositivo captura y procesa, retornando el tipo de gesto o trayectoria ejecutada. En la Tabla 2.2 se describe cada una de las *Clases* contenidas en el API.

CLASES	DESCRIPCIÓN
CircleGesture	Esta clase permite reconocer cuando la punta del dedo dibuja un círculo en el campo de vista del controlador <i>LeapMotion</i> .
Config	Proporciona acceso a la configuración del sistema con métodos para obtener y establecer parámetros de configuración de gestos mediante un objeto <i>Config</i> obtenido a partir de un objeto <i>Controller</i> conectado.

Controller	Interfaz principal del controlador LeapMotion, una instancia de esta clase permite acceder a los frames de datos de seguimiento e información de configuración. Los datos del <i>frame</i> se pueden consultar en cualquier momento.
Device	Esta clase representa un dispositivo conectado físicamente. Además, contiene información relacionada a un dispositivo en particular, como campo de visión, ID de dispositivo y las posiciones calibradas.
DeviceList	Esta clase representa una lista de objetos tipo <i>Device</i> , con métodos que permiten saber si existen dispositivos en conexión.
Finger	La clase representa lo relacionado al seguimiento del dedo, que es clasificado como <i>Finger</i> y es un objeto <i>Pointable</i> en el software LeapMotion. Todas las propiedades rastreadas, tales como la velocidad, seguirán siendo continua en el API.
FingerList	<i>FingerList</i> representa una lista de objetos tipo <i>Finger</i> La clase cuenta con métodos que permiten añadir o eliminar miembros en esta lista.
Frame	Constituye un conjunto de datos de seguimiento de mano y dedos detectados en un solo marco. El software de <i>LeapMotion</i> detecta las manos, los dedos y las herramientas en el área de seguimiento reportan las posiciones, orientaciones, gestos y movimientos en las imágenes a la velocidad de los fotogramas.
Gesture	Detecta el movimiento hecho por el usuario. El controlador observa la actividad dentro de su campo de visión para ciertos patrones de movimientos típicos de gestos de usuario o de comandos. Cuando el software reconoce un gesto, le asigna un ID y agrega un objeto a la lista de gestos.
GestureList	La clase <i>GestureList</i> representa una lista de objetos tipo <i>Gesture</i> .
Hand	Informa las características físicas de la mano detectada, los datos de seguimiento de mano incluye una posición de palma y de la velocidad; vectores para la palma de la mano normal y dirección a los dedos; propiedades de una esfera de ajuste a la mano, las listas de los dedos y las herramientas adjuntas.
HandList	La clase <i>HandList</i> representa una lista de objetos tipo mano.

InteractionBox	Representa una región en forma de caja completamente dentro del campo de vista del controlador <i>LeapMotion</i> . Forma un prisma rectangular alineado con los ejes y proporciona coordenadas normalizadas para las manos, los dedos y las herramientas dentro de esta caja. Esta clase puede hacer más fácil el mapeo de posiciones en el sistema de coordenadas de <i>LeapMotion</i> a sistemas de coordenadas 2D o 3D utilizado para el dibujo en la aplicación.
KeyTapGesture	KeyTapGesture representa un gesto de toque por un dedo o una herramienta. Además, se reconoce cuando la punta de un dedo gira hacia abajo, hacia la palma y luego salta hacia atrás hasta aproximadamente la posición original. El dedo que está tocando debe hacer una breve pausa antes de comenzar el toque.
Listener	Esta clase define un conjunto de funciones <i>callback</i> para responder a eventos distribuidos por un objeto <i>Controller</i> . El objeto <i>Controller</i> llama a las funciones de <i>Listener</i> de un hilo creado por la biblioteca del LeapMotion, no del hilo utilizado para crear o configurar la instancia <i>Listener</i> .
Matrix	La estructura representa una matriz de transformación, se utiliza para transformar <i>vectores</i> , contiene métodos para aplicar transformaciones (como: rotación, escala, igualdad). Las transformaciones pueden ser combinadas por la multiplicación de dos o más matrices de transformación usando el operador <i>*</i> .
Pointable	Maneja las características físicas de un dedo o una herramienta detectada. Los dedos y herramientas se clasifican como objetos orientables (<i>Pointable</i>). La función <i>Pointable</i> se utiliza para determinar si un objeto representa un dedo (<i>isFinger</i>) o la función (<i>ISTool</i>) para determinar si un objeto representa una herramienta. El software de <i>LeapMotion</i> clasifica una entidad detectada como una herramienta cuando es más delgado, recto, y más largo que un dedo típico.
PointableList	La clase <i>PointableList</i> representa una lista de objetos tipo <i>Pointable</i> .
ScreenTapGesture	La clase <i>ScreenTapGesture</i> detecta un gesto de golpe de un dedo o una herramienta. Un gesto de golpe (pulsar) la pantalla se reconoce cuando la punta de un dedo empuja hacia adelante y luego salta hacia atrás

	hasta aproximadamente la posición original, como tocando una pantalla vertical.
SwipeGesture	<i>SwipeGesture</i> es un gesto de deslizar la mano, un dedo o una herramienta. Estos objetos se generan para cada dedo o herramienta visible en la mano que se desliza. Los gestos de desplazamiento son continuos; el objeto aparecerá con el mismo valor de ID en cada cuadro mientras que el gesto continúa.
Tool	El software LeapMotion clasifica una herramienta (<i>Tool</i>) como un objeto <i>Pointable</i> . Las herramientas son objetos más largos, más delgados y más rectos que un dedo típico. Las herramientas pueden hacer referencia a una mano, pero a diferencia de los dedos no están asociados de forma permanente. En cambio, una herramienta puede ser transferida entre las manos mientras se mantiene el mismo ID.
ToolList	La clase <i>ToolList</i> representa una lista de objetos de tipo <i>Tool</i> .
Vector	La estructura <i>Vector</i> representa un vector matemático de tres componentes o punto tal como una dirección o posición en el espacio tridimensional. Los valores dados son en unidades de milímetros del mundo real. El origen se encuentra en el centro del dispositivo.

Tabla 2.2. Descripción clases básicas del API LeapMotion.

2.4. Selección de herramientas Software

Las herramientas software empleadas para el diseño del prototipo, son en su mayoría de código abierto y prestan facilidades para su programación. El desarrollo de la aplicación se basa en el lenguaje de programación JAVA. Se define “*JAVA como un lenguaje de programación puede ser caracterizado como: simple y de alto rendimiento, orientado a objetos, multi-hilo, dinámico, distribuido, seguro, portátil, etc. La ventaja de Java es que la mayoría de los lenguajes de programación o interpreta o compila para ejecutarse en el computador, mientras que Java compila e interpreta*” [26].

En los últimos años, Java ha mejorado considerablemente, sobre todo desde la aparición del estándar Java 2. Ahora, bajo este lenguaje de programación, hay muchas oportunidades para crear aplicaciones con interfaz de usuario gráfica (GUI), un procesamiento de imágenes y programación detallada de los elementos gráficos.

JAVA presenta múltiples enfoques de programación que dependen del tipo de entorno de funcionamiento y uno de estos enfoques es JAVA3D; *“Es una interfaz de programación de aplicaciones (API), desarrollado en Sun Microsystems para la representación de gráficos 3D interactivos utilizando el lenguaje de programación Java. Java 3D es una API de Java del tipo client-side³”* [27].

Por otro lado, es importante el uso del estándar UML (Lenguaje Unificado de Modelado) que se aplica al enfoque orientado a objetos y ofrece vistas apropiadas del sistema. Brinda la posibilidad de visualización en múltiples dimensiones y niveles, es adecuado para determinar la estructura y el comportamiento del programa. En general, esta es una buena manera de resolver los problemas que se presentan al diseñar software. El UML describe el código fuente y los modelos ayudan a visualizar el sistema. Por lo tanto, determina lo que es o lo que debería ser.

2.4.1. Metodología de desarrollo de software

En el presente trabajo de grado se requiere de una metodología de desarrollo de software que permita integrar el entorno de desarrollo *NetBeans*, *MySQL* como el sistema de gestión de bases de datos relacional para el almacenamiento y administración de datos y las APIs del dispositivo sensor *LeapMotion*.

“La ingeniería del software establece y se vale de una serie de modelos que establecen y muestran las distintas etapas y estados por los que pasa un producto software, desde su concepción inicial, pasando por su desarrollo, puesta en marcha y posterior mantenimiento, hasta la retirada del producto” [28].

De lo anterior, se puede decir que la metodología de desarrollo de software permite visualizar las actividades que ocurren durante el desarrollo del mismo. De igual manera, intenta determinar el orden de las etapas involucradas y los criterios de transición asociados entre estas etapas. Sin embargo, existen diferentes metodologías y una elección acertada de esta es realmente importante porque de ella depende el éxito o fracaso del proyecto software.

A continuación, en la Tabla 2.3 se muestran algunos de las metodologías tradicionales y más utilizadas [28].

³La definición Client-side refiere a las operaciones realizadas por el cliente en una relación cliente/servidor. Por lo tanto, el cliente es una aplicación de computador que se ejecuta de forma local y se conecta a un servidor.

METODOLOGÍA	CARACTERÍSTICA	VENTAJA	DESVENTAJA
Cascada	<p>Es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior y es un proceso de desarrollo secuencial.</p>	<ul style="list-style-type: none"> • Es apropiado especialmente para proyectos con requisitos no cambiantes. Donde es posible y probable que los diseñadores predigan los problemas del sistema y produzcan un diseño correcto antes de la implementación. • Es un modelo en el que todo está bien organizado y no se mezclan las fases. Es simple y fácil de usar. • Debido a la rigidez del modelo es fácil de gestionar ya que cada fase tiene entregables específicos y un proceso de revisión. 	<ul style="list-style-type: none"> • En la práctica, un proyecto rara vez sigue una secuencia lineal, esto crea una mala implementación del modelo, lo cual hace que lo lleve al fracaso. • Difícilmente un cliente va a establecer al principio todos los requisitos necesarios, por lo que provoca un gran atraso trabajando en este modelo, ya que este es muy restrictivo y no permite movilizarse entre fases. • Se considera un modelo pobre para proyectos complejos, largos, orientados a objetos y por supuesto en aquellos en los que los requisitos tengan un riesgo de moderado a alto de cambiar. Genera altas cantidades de riesgos e incertidumbres.

<p style="text-align: center;">En V</p>	<p>El modelo en V significa validación y verificación, la forma de la V muestra que en la parte izquierda de ella representa la descomposición de los requisitos y la creación de las especificaciones del sistema. El lado derecho de la v representa la integración de partes y su verificación.de desarrolló. Además, en el modelo las pruebas necesitan empezarse lo más pronto posible en el ciclo de vida.</p>	<ul style="list-style-type: none"> • Es un modelo simple y fácil de utilizar. • En cada una de las fases hay entregables específicos. • Tiene una alta oportunidad de éxito sobre el modelo en cascada debido al desarrollo de planes de prueba en etapas tempranas del ciclo de vida. • Es un modelo que suele funcionar bien para proyectos pequeños donde los requisitos son entendidos fácilmente. 	<ul style="list-style-type: none"> • Es un modelo muy rígido, como el modelo en cascada. • Tiene poca flexibilidad y ajustar el alcance es difícil y caro. • El software se desarrolla durante la fase de implementación, por lo que no se producen prototipos del software. • El modelo no proporciona caminos claros para problemas encontrados durante las fases de pruebas
<p style="text-align: center;">Iterativo</p>	<p>Este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de recogida de requisitos. Algunas veces este modelo es usado en proyectos en los que los requisitos no están claros por parte del usuario. Además, consiste en la iteración de varios ciclos de vida en cascada y al final de cada iteración se entrega al cliente una versión mejorada del producto. El cliente es el que evalúa, corrige el producto o propone mejoras.</p>	<ul style="list-style-type: none"> • No hace falta que los requisitos estén totalmente definidos al inicio del desarrollo, sino que se pueden ir refinando en cada una de las iteraciones. • Realiza el desarrollo en pequeños ciclos, lo que permite gestionar mejor los riesgos y gestionar mejor las entregas. 	<ul style="list-style-type: none"> • El no tener los requisitos definidos desde el principio, puede verse también como un inconveniente ya que pueden surgir problemas relacionados con la arquitectura.

<p>De desarrollo incremental</p>	<p>Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software.</p>	<ul style="list-style-type: none"> • Mediante este modelo se genera software operativo de forma rápida y en etapas tempranas del ciclo de vida del software. • Es un modelo más flexible, por lo que se reduce el costo en el cambio de alcance y requisitos. • Es más fácil probar y depurar en una iteración más pequeña. • Es más fácil gestionar riesgos. • Cada iteración es un hito gestionado fácilmente 	<ul style="list-style-type: none"> • Para el uso de este modelo se requiere una experiencia importante para definir los incrementos y distribuir en ellos las tareas de forma proporcionada. • Cada fase de una iteración es rígida y no se superponen con otras. • Pueden surgir problemas referidos a la arquitectura del sistema porque no todos los requisitos se han reunido, ya que se supone que todos ellos se han definido al inicio.
<p>Espiral</p>	<p>Tiene en cuenta fuertemente el riesgo que aparece a la hora de desarrollar software. Para ello, se comienza mirando las posibles alternativas de desarrollo, se opta por la de riesgos más asumibles y se hace un ciclo de la espiral. Si el cliente quiere seguir haciendo mejoras en el software, se vuelven a evaluar las nuevas alternativas y riesgos y se realiza otra vuelta de la espiral, así hasta que llegue un momento en el que el producto software desarrollado sea aceptado.</p>	<ul style="list-style-type: none"> • Reduce riesgos del proyecto • Incorpora objetivos de calidad. • Integra el desarrollo con el mantenimiento. • Es posible tener en cuenta mejoras y nuevos requerimientos sin romper con el modelo, ya que el ciclo de vida no es rígido ni estático. 	<ul style="list-style-type: none"> • Es un modelo que genera mucho trabajo adicional. Al ser el análisis de riesgos una de las tareas principales exige un alto nivel de experiencia y cierta habilidad en los analistas de riesgos (es bastante difícil). • Es un modelo costoso. Además, no es un modelo que funcione bien para proyectos pequeños.
<p>De Prototipos</p>	<p>Construcción de prototipos</p>	<ul style="list-style-type: none"> • Ofrece visibilidad 	<ul style="list-style-type: none"> • Puede ser un

<p>De Prototipos</p>	<p>comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un diseño rápido. El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente. El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.</p>	<p>del producto desde el inicio del ciclo de vida con el primer prototipo. Esto puede ayudar al cliente a definir mejor los requisitos y a ver las necesidades reales del producto.</p> <ul style="list-style-type: none"> • Permite introducir cambios en las iteraciones siguientes del ciclo y la realimentación continua del cliente. • Este modelo reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios. 	<p>desarrollo lento. Además se hacen fuertes inversiones en un producto desechable ya que los prototipos se descartan. Esto puede hacer que aumente el costo de desarrollo del producto.</p> <ul style="list-style-type: none"> • Pueden surgir problemas con el cliente que ve funcionando versiones del prototipo pero puede desilusionarse porque el producto final aún no ha sido construido. • El desarrollador puede caer en la tentación de ampliar el prototipo para construir el sistema final sin tener en cuenta los compromisos de calidad y de mantenimiento que tiene con el cliente.
<p>Extreme Programming (XP)</p>	<p>Surge como una nueva manera de encarar proyectos de software, proponiendo una metodología basada esencialmente en la simplicidad y agilidad. Las metodologías de desarrollo de software tradicionales (ciclo de vida en cascada, evolutivo, en espiral,</p>	<ul style="list-style-type: none"> • XP propone un ciclo de vida dinámico, ya que fomenta la comunicación entre el cliente y el programador, facilitando así los cambios al 	<ul style="list-style-type: none"> • Es recomendable usarlo solo en proyectos a corto plazo. • Algunas veces su implementación puede que no sea más sencilla que el desarrollo de

<p style="text-align: center;"><i>Extreme Programming (XP)</i></p>	<p>iterativo, etc.) aparecen, comparados con los nuevos métodos propuestos en XP, como pesados y poco eficientes. XP es una metodología ágil de desarrollo de software más exitosa de los últimos tiempos. Además, está diseñada para entregar el software que los clientes necesitan en el momento en que lo necesitan. XP permite responder a los requerimientos cambiantes de los clientes, aún en fases tardías del ciclo de vida del desarrollo [29]</p>	<p>producto y ocasionando eficiencia en el proceso de planificación y pruebas.</p> <ul style="list-style-type: none"> • Trata de realizar ciclos de desarrollo cortos (llamados iteraciones), con entregables funcionales al finalizar cada ciclo. • Permite ahorrar tiempo y dinero. • Da la posibilidad de tener una programación más organizada. 	<p>metodologías tradicionales.</p> <ul style="list-style-type: none"> • Requiere de un rígido ajuste a los principios de XP.
---	---	--	---

Tabla 2.3 Metodologías de desarrollo de software tradicionales.

Después de realizar un análisis comparativo por medio de la tabla anterior, se puede definir que la metodología de desarrollo de software a usar en el presente trabajo de grado es la metodología ***Extreme Programming (XP)***. Ya que, se adapta a las necesidades, tiempo y costo del proyecto.

CAPÍTULO 3

TRAYECTORIAS

Para tratar los trastornos de motricidad fina en miembro superior, se realizan ejercicios de rehabilitación que involucran el seguimiento de trayectorias. Dichas trayectorias pueden ser elaboradas en plástico, metal o madera, con el propósito de ser recorridas con un objeto que se toma con la mano o pueden estar dibujadas sobre papel, como por ejemplo laberintos para ser seguidos mediante el uso de un bolígrafo.

Además, existen una gran variedad de ejercicios para rehabilitación, el más común consiste en hacer seguimiento a una trayectoria recta hecha de un material conductor con un aro metálico guiado por la mano del paciente. El propósito de este ejercicio es sujetar y desplazar el aro por la trayectoria sin tocarla; si el aro y la trayectoria hacen contacto, se cierra un circuito que enciende una luz que indica que el ejercicio no se está realizando correctamente.

En el presente trabajo de grado se implementa un ambiente virtual para rehabilitación de motricidad fina en miembro superior, en el cual se desarrolla un ejercicio de seguimiento de trayectoria, que a diferencia del desarrollado en el trabajo de grado precedente [30], no necesita de un dispositivo con capacidad de retorno de fuerza.

Al trabajar con la asistencia del campo visual, el paciente puede realizar ejercicios que impliquen trayectorias curvas de diferente apariencia, distintas a la trayectoria común que se toma en el ejercicio real. El dispositivo *LeapMotion* utiliza un sistema de coordenadas cartesiano diestro (Ejes X, Y, Z), donde el origen se encuentra en la parte superior del dispositivo, esto se puede lograr debido a que tiene un campo de rotación de 360 grados sobre el espacio.

En este capítulo se describe el trazado de trayectorias en 2D y 3D. Así como, el método adoptado para el modelado de las trayectorias utilizadas en el ambiente virtual.

3.1. Trazado de trayectorias en 2D

Para el trazado de las trayectorias en el ambiente virtual es necesario describirlas matemáticamente, con el fin de conocer las coordenadas cartesianas de cada punto para hacer el control de seguimiento con el dispositivo.

Teniendo en cuenta lo expuesto en [30], *“Una trayectoria curva se puede representar mediante un conjunto de segmentos rectos; y la aproximación a la forma requerida es mucho mejor cuanto mayor sea el número de segmentos. El principal problema de este tipo de aproximación es la gran cantidad de coordenadas de cada punto inicial y final de los segmentos, que se deben almacenar para lograr una aproximación razonable. Un problema adicional, es que se deben posicionar correctamente dichos puntos de forma interactiva.”*

En [30], se presentaron dos métodos para trazar trayectorias espaciales. El primero de ellos basado en funciones polinomiales y el segundo basado en las curvas de *Bézier*. Debido a su simplicidad y flexibilidad se seleccionó el segundo método, el cual consiste en: *“conocer cierta cantidad de puntos en el espacio o en el plano, dependiendo del tipo de curva que se desee generar, espacial o planar. Estos puntos se denominan vértices de control, porque con ellos se construye el polígono de control; el cual, es una trayectoria trazada a través de los vértices y sirve para definir el comportamiento de la curva. Una mirada al polígono de control puede brindar información aproximada sobre la apariencia que tendrá la curva de Bézier. El orden de una curva de Bézier se define por el número de vértices de control menos uno, que se hayan utilizado para generarla.”*

El método de las curvas de *Bézier* utiliza el *polinomio de Bernstein*, cuya expresión se muestra en la ecuación (3.1)

$$Bi^N = \binom{N}{i} (1 - s)^{N-i} s^i \quad (3.1)$$

$$\binom{N}{i} = \frac{N!}{i! (N - i)!} \quad (3.2)$$

La ecuación (3.2) muestra el coeficiente binomial del Teorema del Binomio de Newton donde N es el orden de la curva. Además, en la fórmula de las curvas de *Bézier* se incluyen los valores de los vértices de control P_i , y como resultado

se obtiene la formula (3.3), donde los valores que toma la variable s , se encuentra en el intervalo $[0,1]$.

$$B(s) = \sum_{i=0}^N B_i^N(s) P_i \quad s \in [0,1] \quad (3.3)$$

Para obtener la fórmula de una curva de *Bézier* lineal (3.4), partiendo de (3.3), con puntos de control P_0 y P_1 en R^3 , se requiere del proceso descrito a continuación:

$$P_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad , \quad P_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

$$B(s) = B(s)_0 P_0 + B(s)_1 P_1$$

$$B(s)_0 P_0 = (1 - s) \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

$$B(s)_1 P_1 = s \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

$$B(s) = (1 - s) \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + s \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad s \in [0,1] \quad (3.4)$$

En este caso, la curva de *Bézier* lineal contiene dos puntos de control, P_0 y P_1 , la variable s describe la distancia en la que se encuentra $B(s)$ en su trayecto de P_0 hacia P_1 . Así como se muestra en la figura siguiente (Figura 3.1).

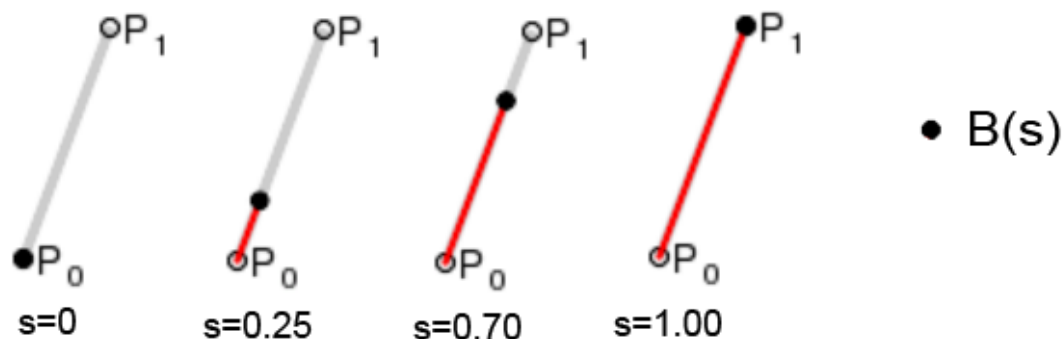


Figura 3.1. Trayecto de $B(s)$, curva de Bézier de primer orden. Tomado de: Animated linear Bézier curve (Philip Tregoning)

La Figura 3.1, muestra el comportamiento de la variable s en el recorrido de P_0 hacia P_1 . Se puede ver que la variable s toma valores en el intervalo cerrado $[0,1]$; cuando $s = 0.25$ el vector $B(s) - P_0$ tiene una magnitud igual a un cuarto de la magnitud del vector $P_1 - P_0$.

En cambio, en una curva de orden dos se necesitan tres puntos de control (P_0, P_1 y P_2). Además, para trazar la trayectoria de $B(s)$ se toman las dos rectas que forman parte del polígono de control; la primera recta se origina en el punto P_0 y va hasta P_1 y la segunda recta va del punto P_1 a P_2 . Cuando la variable s va tomando valores en el intervalo $[0,1]$, existe un punto Q_0 que se mueve sobre la primer recta y un punto Q_1 que se desplaza por la segunda recta del polígono de control, como una curva de Bézier de primer orden entre los puntos Q_0 y Q_1 anteriormente descritos. Por lo tanto, el trayecto de $B(s)$ se construye desde el punto Q_0 hasta Q_1 formando una curva cuadrática. Así como se muestra a en la Figura 3.2

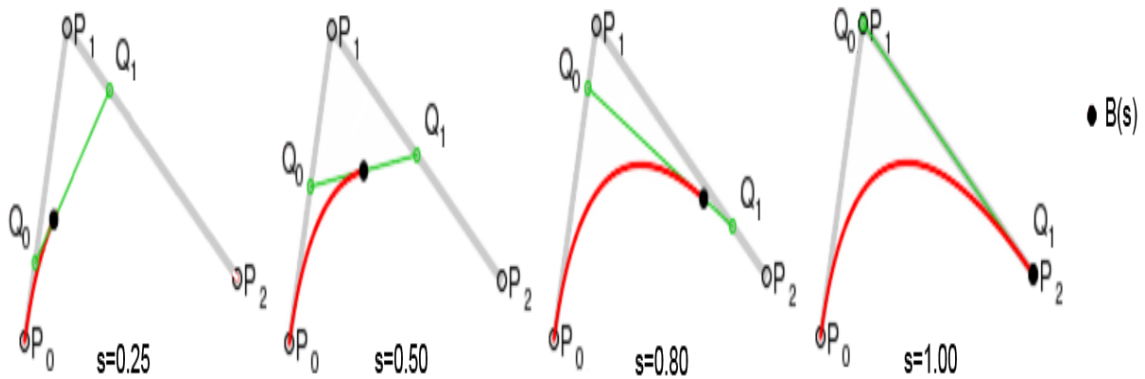


Figura 3.2. Trayecto de $B(s)$, curva de Bézier de segundo orden. Tomado de: Animated quadratic Bézier curve (Philip Tregoning)

En conclusión, para construir una curva de Bézier de orden n , es necesario trazar el polígono de control teniendo en cuenta los “ $n+1$ ” puntos de control necesarios para trazar dicha curva. Además, se debe localizar los puntos intermedios que van aumentando de orden desde el polígono de control hasta que se adquiera el comportamiento de $B(s)$ en el orden deseado. Para explicar el comportamiento de una curva de Bézier de mayor orden, se toma como ejemplo una curva de cuarto orden. Así como se observa en la Figura 3.3

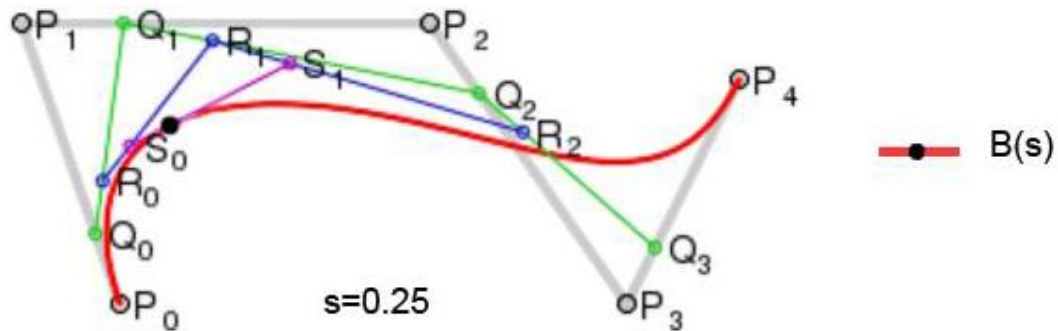


Figura 3.3. Trayecto de $B(s)$, curva de Bézier de cuarto orden. Tomado de: Animated fourth order Bézier curve (Philip Tregoning)

Se observa en la Figura 3.3, el polígono de control está conformado por cuatro segmentos de recta que son: $\overline{P_0P_1}, \overline{P_1P_2}, \overline{P_2P_3}, \overline{P_3P_4}$. Además, están los puntos intermedios (Q_0, Q_1, Q_2 y Q_3) que describen el movimiento lineal de la curva; Y sobre estos puntos se ubican los puntos R_0, R_1 y R_2 que describen trayectorias

cuadráticas. Finalmente, la curva de Bézier de cuarto orden $B(s)$ se desplaza entre los puntos S_0 y S_1 .

3.2. Trayectorias en 3D del ambiente virtual aplicando el método de curvas de Bézier.

Para construir las trayectorias del ambiente virtual se utilizó el método de curvas de Bézier, esto con el fin de tener diferentes trayectorias de forma aleatoria para el entrenamiento del ambiente virtual. De acuerdo, con el trabajo de grado realizado en [30], en este determinan el número de vértices de control que deben tener las trayectorias para un dispositivo háptico con retroalimentación de campo fuerza, concluyendo que: *“una cantidad alta de vértices de control son la complejidad de la trayectoria resultante para ser seguida por el paciente y el consumo de recursos computacionales si se tiene en cuenta que el sistema debe presentar respuestas en tiempo real de ejecución.”*

Considerando lo anterior, el dispositivo LeapMotion no está limitado por curvas cerradas en la trayectoria debidas a la cantidad de vértices de control; logrando que el paciente pueda seguir la trayectoria sin ningún inconveniente. Ya que, el dispositivo tiene un campo de rotación de 360 grados sobre el espacio.

Además, las trayectorias no son precargadas ya que se generan aleatoriamente los puntos de control con la elección de nivel de complejidad por parte del usuario, el nivel de complejidad se puede elegir como un valor entero entre 1 a 10. Para lograr el renderizado de la trayectoria de Bézier a través de los puntos de control se utilizan primitivas de las librerías de JAVA 3D.

Como se dijo anteriormente, la ecuación (3.3) refleja matemáticamente el trazado de una curva de Bézier; en el presente trabajo de grado se tomó dicha ecuación para determinar la trayectoria a seguir por el paciente con la variación de que esta es tridimensional. Para lograr este efecto en la trayectoria se utilizaron coordenadas esféricas para generar los puntos de control de Bézier. Los vértices de control determinan la posición espacial de un punto mediante una magnitud y dos ángulos, así como se muestra en la Figura 3.4. Además, se usan las coordenadas esféricas para posicionar dichos vértices o puntos de control para generar la trayectoria y de este modo ser implementado en el ambiente virtual.

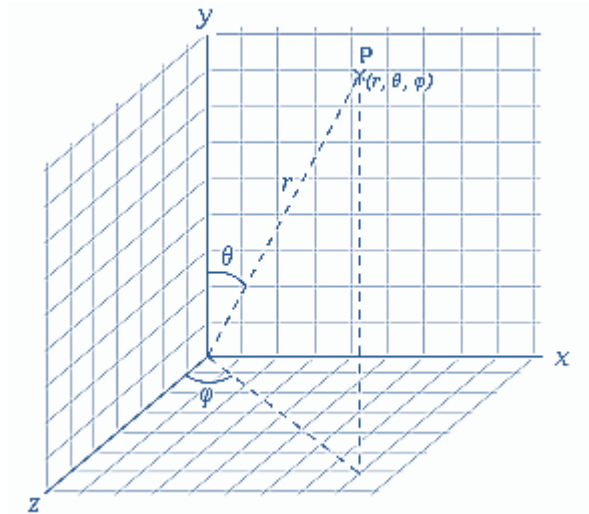


Figura 3.4. Representación espacial del vértice de control P usando coordenadas esféricas. Fuente propia

En la Figura 3.4, se puede observar que el punto P está representado por tres magnitudes, que son: r (radio), θ (ángulo polar o latitud) y φ (azimut). Además, θ está comprendido desde 0° a 180° (en radianes de 0 a π) y cero sobre la parte positiva del eje Y. Por otro lado, la azimut varía de acuerdo a la medida del ángulo φ , este puede ser en el sentido de las manecillas del reloj o contra reloj, este ángulo varía entre 0 y $\pm\pi$

3.3. Modelo matemático para el renderizado de la trayectoria 3D del ambiente virtual

Para generar el renderizado de la trayectoria en 3D se utilizó un modelo matemático que se basa en la construcción de vectores por medio de los puntos de la curva de Bézier. Teniendo en cuenta la Figura 3.3, $B(s)$ se desplaza entre los puntos S_0 y S_1 , entre los cuales se genera un vector director (\vec{V}_d) que va de S_0 a S_1 . Además, se requiere calcular el punto medio (P_m) entre el segmento $\overline{S_0S_1}$. Para el cálculo de \vec{V}_d se utiliza la ecuación (3.5) y para el punto medio la ecuación (3.6), como se observa en la Figura 3.5.

$$\vec{V}_d = S_1 - S_0 \quad (3.5)$$

$$P_m = \frac{S_1 + S_0}{2} \quad (3.6)$$

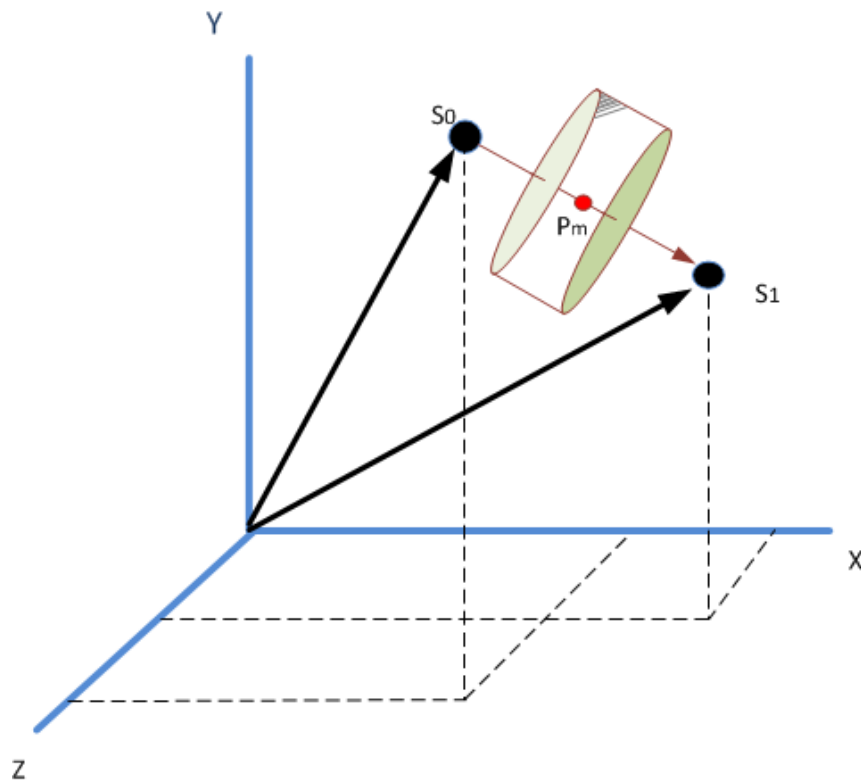


Figura 3.5. Posicionamiento del cilindro en el segmento generado entre S_0 y S_1 . Fuente propia

De acuerdo al modelo, el renderizado consiste en posicionar un cilindro en el punto medio en dirección de \vec{V}_d para cada punto de la curva. Los incrementos de s , $s \in [0,1]$, son tan pequeños que genera el efecto de unir cada cilindro dentro de la curva y así conformar una trayectoria 3D. La ubicación del cilindro dentro de la trayectoria conlleva una transformación de las coordenadas del vector director para obtener los ángulos de rotación (θ y φ). El cilindro posee un eje principal con centro de coordenadas (r, φ, θ) , con $\theta = 0$. Se traslada el cilindro a la posición de cada punto medio y se rota en la dirección del vector director; dando como resultado el desplazamiento del cilindro en la curva. Así como, se muestra en la Figura 3.6.

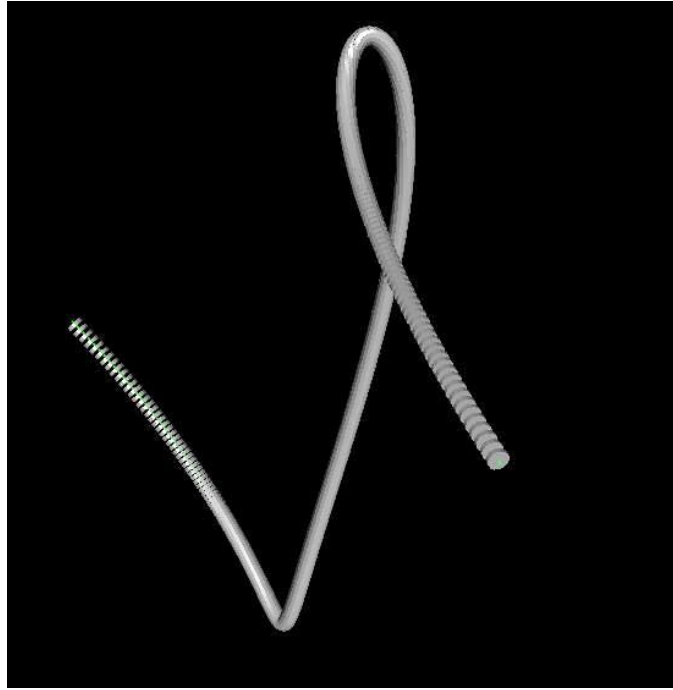


Figura 3.6. Representación de la trayectoria en 3D en el ambiente virtual. Fuente propia

3.4. Cálculo de la orientación del Toroide

En esta sección se presenta el procedimiento de cálculo utilizado para determinar la orientación del toroide que representa el anillo que guía el usuario. A continuación, en la ecuación (3.7) se expresa el modelo matemático usado.

$$\begin{array}{ccc}
 I = \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} & P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} & D = \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} & (3.7) \\
 \text{Posición del índice} & \text{Posición del pulgar} & \text{Dirección en que apunta el dedo Índice} &
 \end{array}$$

El vector I es generado a partir de la posición del dedo índice y el vector P se origina de la posición del dedo pulgar de la mano, estos vectores son capturados por el dispositivo LeapMotion que determina cada punto y su respectiva posición en el espacio. El vector \vec{D} está orientado en la dirección del dedo índice, como se puede observar en la siguiente figura.

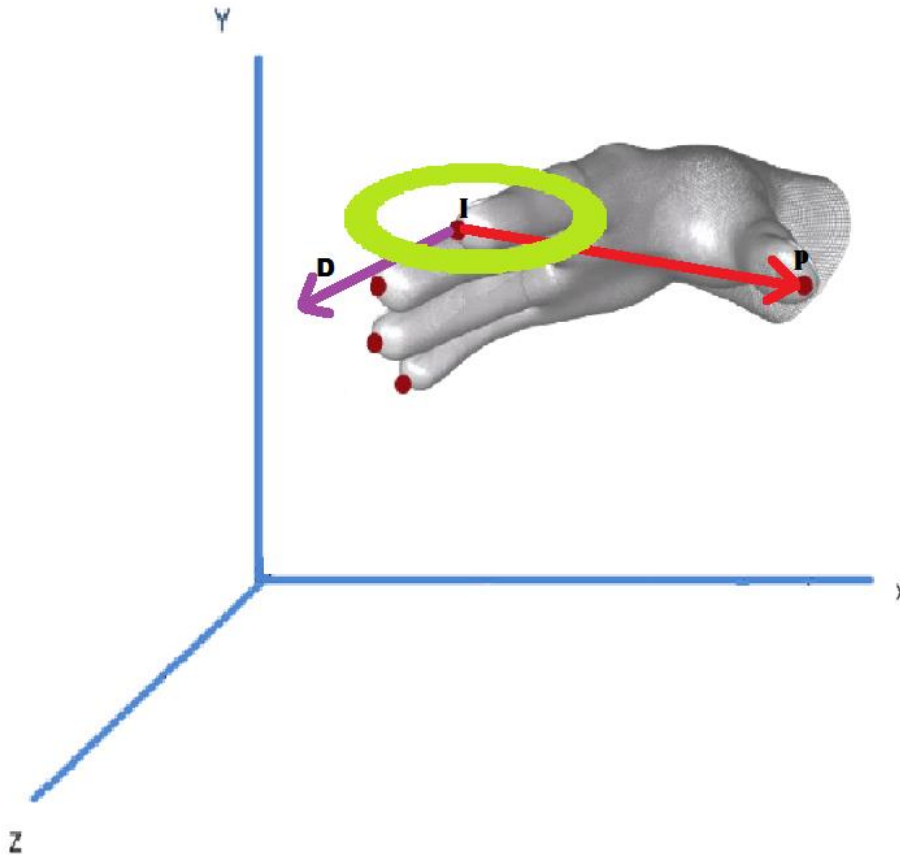


Figura 3.7. Orientación del Toroide. Fuente propia

De la Figura 3.7, se puede observar que entre los puntos P e I se genera un vector \vec{IP} y entre el vector \vec{IP} y el vector \vec{D} se genera un plano; sobre dicho plano se encuentra posicionado el toroide. Para la ubicación del toroide, primero se realiza el desplazamiento de este en el espacio hasta el punto deseado. Para orientarlo se calcula el ángulo θ que está ubicado entre el vector \vec{D} y el eje positivo Y , el ángulo φ está entre la proyección del vector director del dedo índice (\vec{D}) sobre el plano xz y el eje positivo de Z . Además, el tercer ángulo α se encuentra entre el vector de la proyección de \vec{IP} sobre el plano yz y el eje positivo Z , Así como, se muestra en la Figura 3.8. Las rotaciones se hacen en un orden sucesivo al plano xyz , iniciando con φ que es la rotación que se le da al toroide alrededor del eje y , el ángulo θ gira sobre el eje z , finalmente α alrededor del eje x .

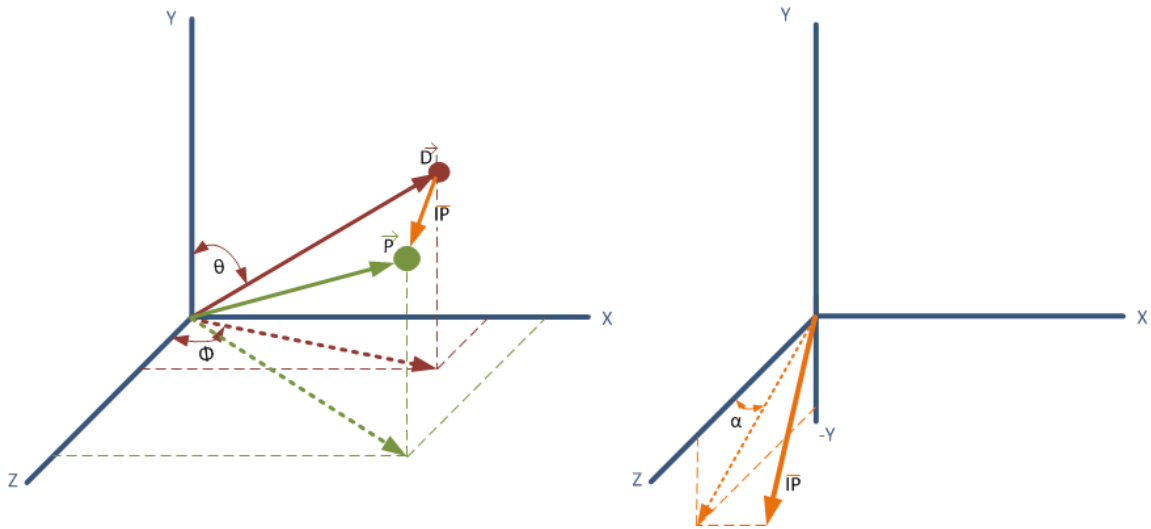


Figura 3.8. Posición y Dirección del Toroide. Fuente propia

El ambiente virtual también cuenta con una esfera de la cual cabe aclarar que es simplemente un punto ubicado en el espacio, que se posiciona con respecto al el vector I (posición de dedo índice) y este a su vez es detectado por el dispositivo que determina la posición de dicho punto en el espacio.

3.5. Modelo matemático para el renderizado del toroide del ambiente virtual

Para generar el renderizado del toroide se utilizó un modelo matemático que se basa en la construcción rectángulos conformados por 4 puntos en el espacio, estos puntos son calculados por medio de desplazamientos de los ángulos β y α por medio de la ecuación (3.8); donde α es la latitud del punto respecto del plano XY, y β el ángulo de rotación de la circunferencia generatriz alrededor del eje Y. Tal como, se tiene en la Figura 3.9; haciendo desplazamientos sucesivos en la dirección de α para ubicar cada rectángulo y el siguiente desplazamiento de β otro rectángulo el cual está dividido en 40 x40 secciones. Así como se muestra en la Figura 3.10.

$$\begin{cases} x = (R + r \cos \alpha) \cos \beta \\ y = (R + r \cos \beta) \sin \beta \\ z = r \sin \alpha \end{cases} \quad (3.8)$$

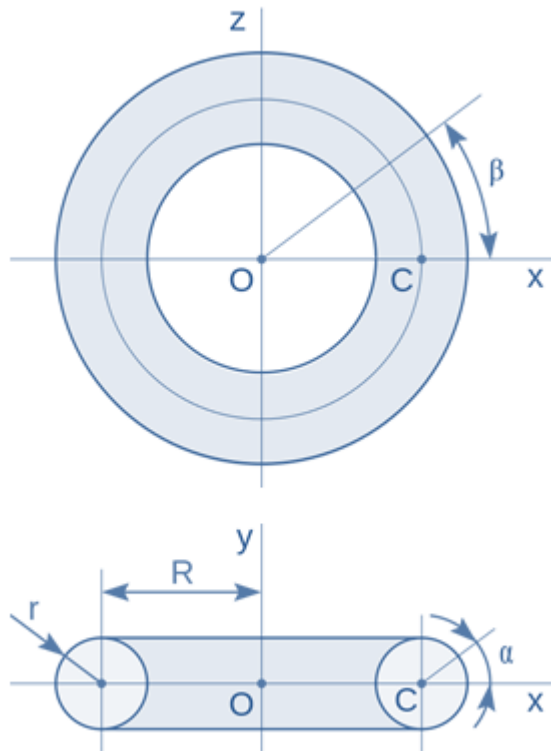


Figura 3.9. Desplazamientos de los ángulos β y α para la construcción del toroide. Tomado de Wikipedia

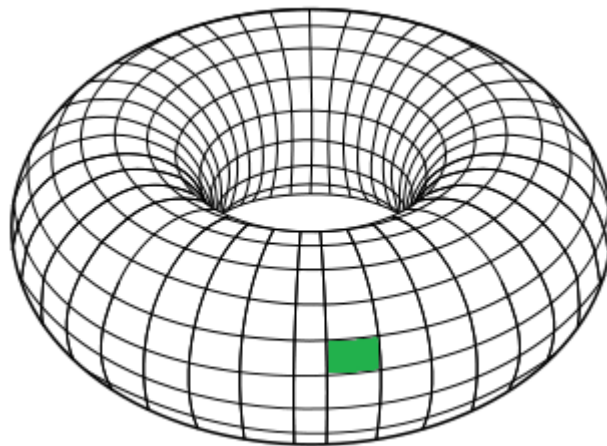


Figura 3.10. Ubicación de cada rectángulo en el volumen del toroide. Tomado de Wikipedia

CAPÍTULO 4

DESCRIPCIÓN DE LA INTERFAZ GRÁFICA Y AMBIENTE VIRTUAL DEL PROTOTIPO DESARROLLADO

El avance y consolidación de las herramientas computacionales han facilitado el desarrollo de aplicaciones software con propósitos de rehabilitación motriz en dos aspectos fundamentales: uno es el almacenamiento y administración de datos; y otro es proporcionar medios que permitan realizar actividades propias del proceso de rehabilitación. El primer aspecto se beneficia de la ya madura tecnología de bases de datos y el segundo de las nuevas tecnologías como realidad virtual y visión computacional.

En este capítulo se realiza una descripción de las características de funcionamiento del prototipo desarrollado en este trabajo de grado. Además, el prototipo se constituye de la interfaz de usuario y del ambiente virtual. Dentro de la interfaz de usuario existen tres formularios; en el primero se registra la información del paciente donde se puede ingresar sus datos personales. El segundo formulario permite realizar la evaluación del paciente contestando una serie de preguntas relacionadas con la destreza del miembro superior del usuario y cuenta con un campo destinado para que el especialista a cargo consigne las observaciones acerca del estado del paciente. El tercer formulario permite realizar consultas sobre el paciente como: sus datos personales, resultados de las evaluaciones que se le han realizado y resultado de desempeño en la realización de los ejercicios en el ambiente virtual. Dentro del formulario se pueden listar los registros de pacientes; también se puede realizar la consulta de los datos de un paciente, esta consulta se hace a través de su número de cédula o su nombre. Por otro parte, para el ambiente virtual se implementan dos opciones; la primera es un acceso a un DEMO que permite al usuario entender cómo funcionan los ejercicios de entrenamiento y la segunda es el ambiente virtual donde se cargan los ejercicios de entrenamiento, cada uno con una trayectoria diferente. En el **ANEXO B**, se muestra el procedimiento de instalación de la aplicación con todos sus requerimientos.

De acuerdo a lo mencionado en el capítulo 2, se tiene en cuenta la aplicación de la metodología de desarrollo de software XP seleccionada en dicho capítulo. Por lo tanto, con esta se trata de realizar ciclos de desarrollo o iteración cortos

con entregables funcionales al final de cada ciclo. En cada iteración se realiza un ciclo completo planificación, diseño, desarrollo y pruebas. Así como se muestra en la Figura 4.1

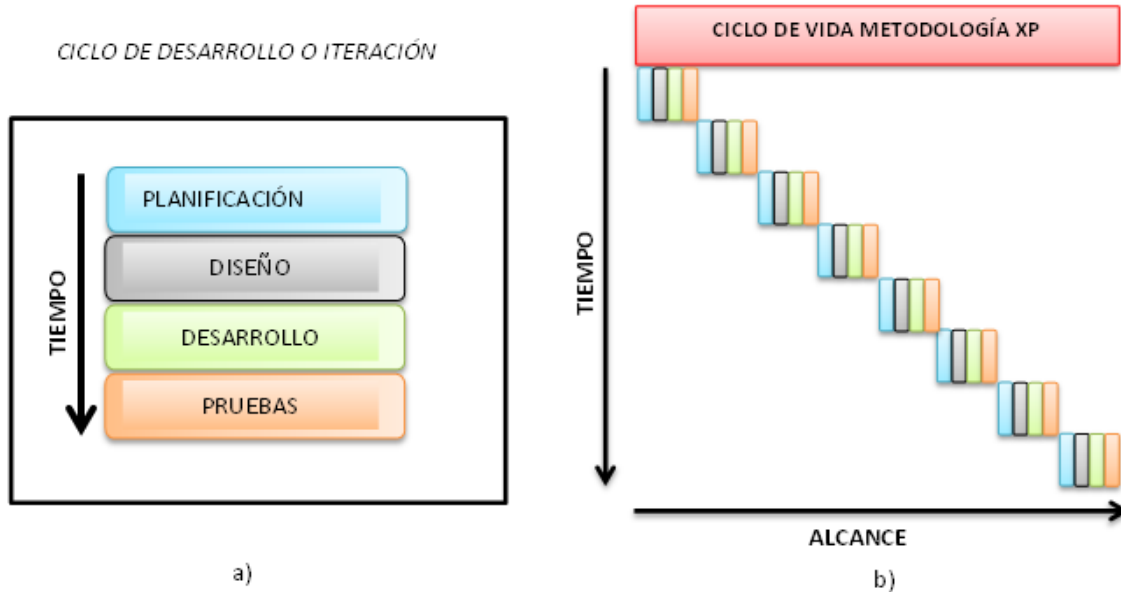


Figura 4.1. Metodología XP. a) Ciclo de desarrollo o iteración y b) Ciclo de vida metodología XP. Tomado de [29]

1.1. Descripción de la Interfaz gráfica de Usuario

Teniendo en cuenta el ciclo de desarrollo o iteración de la metodología XP [29] y de acuerdo a las fases contenidas en él (planificación, diseño, desarrollo y pruebas), la interfaz de usuario se desarrolla así como se describe a continuación:

1.1.1. Ciclo de desarrollo de la interfaz gráfica de usuario:

Se inicia en la etapa de planificación, donde se recopila la información requerida para posteriormente realizar el diseño, desarrollo y pruebas de la interfaz. Como tal, la fase de planificación debe contener las historias de usuario que es la descripción detallada de lo que el cliente desea, en la fase de diseño se realiza el modelado en UML para determinar la relación existente entre historias de usuario, finalmente en la fase de desarrollo y pruebas se presenta la implementación de los formularios y el desempeño de la interfaz.

1.1.1.1. Fase de planificación

De acuerdo a lo anterior, y teniendo en cuenta los requerimientos del proyecto desarrollado en este trabajo de grado a continuación se detallan las historias de usuario.

Historias de Usuarios

Se desea crear un ambiente virtual para rehabilitación motriz fina de miembro superior con el fin de establecer un control de seguimiento de una trayectoria en 3 dimensiones, mediante el dispositivo *LeapMotion*, permitiendo que el usuario interactúe con el ambiente virtual sin necesidad de tocar el dispositivo.

Además, la interfaz gráfica debe ser amigable y entendible, tanto para el profesional a cargo como para el paciente, esta interfaz debe realizar las siguientes funciones:

1. *Registro Pacientes*: Debe registrar los datos personales del paciente como: nombre y apellido, tipo y número de documento, género, edad, ciudad de origen, número telefónico, fecha en la que se realiza el registro y evaluador o profesional a cargo. Por otro lado, las opciones para buscar, guardar, eliminar y limpiar deben estar incluidas en esta interfaz.
2. *Consulta Pacientes*: Por medio del nombre o número de documento debe encontrar el registro del paciente en la base de datos del sistema. Además, debe contener las opciones de: Resultados datos del paciente, Resultado evaluación pacientes y Resultado desempeño pacientes. En el primer ítem debe listar las opciones encontradas que coincidan con el registro buscado, en el segundo debe mostrar los resultados de la evaluación del paciente encontrado y en el último ítem se deben ver los resultados de desempeño del usuario en mención.
3. *Práctica y Ejercicios*: Está debe dirigir al usuario al ambiente virtual, donde se debe encontrar las opciones de la rehabilitación y un ejercicio de práctica para que el paciente se acople e interactúe con la realidad virtual.

Cabe resaltar, las funciones descritas anteriormente también son historias de usuarios

1.1.1.2. Fase de diseño

Para entender mejor cómo funciona el sistema y su relación entre historias de usuarios o casos de uso, se tienen en cuenta el uso de modelos que sirven para explicar de manera simplificada la interfaz gráfica de usuario desarrollada en este trabajo de grado. Así como se muestra en la Figura 4.2 el diagrama de casos de uso de la interfaz gráfica de usuario.

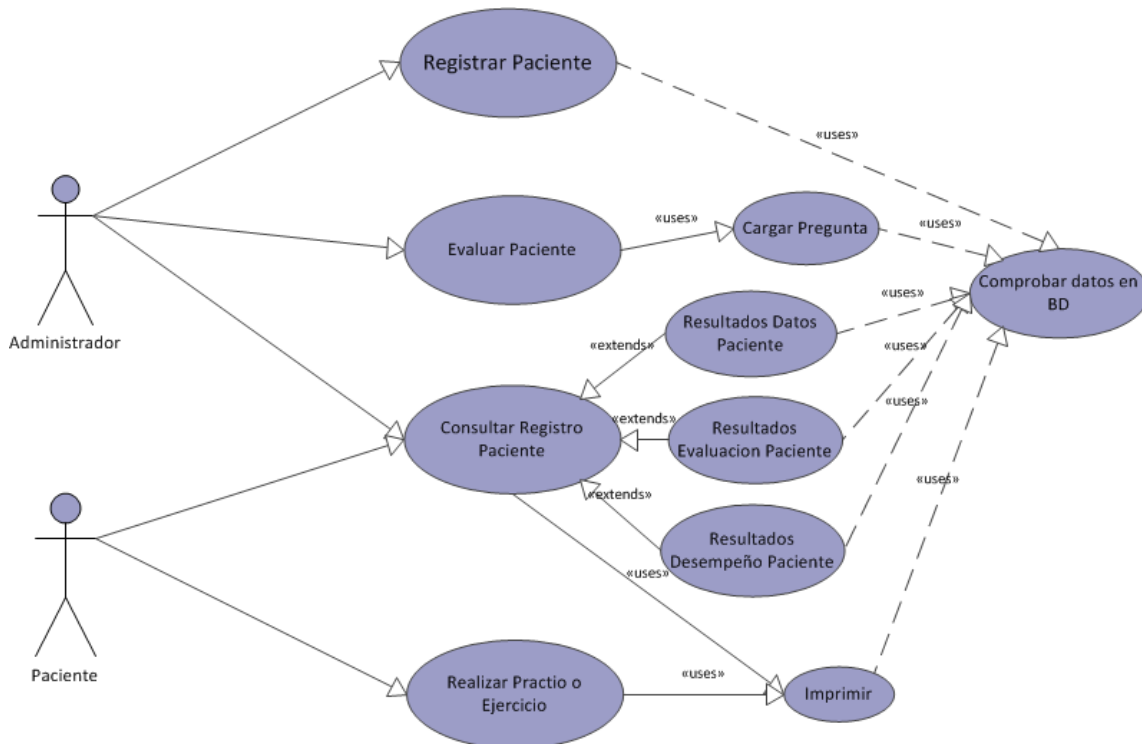


Figura 4.2. Diagrama de casos de uso del sistema. Fuente propia

De la anterior gráfica, se puede ver el sistema desde la perspectiva de sus usuarios (Administrador y Paciente) y así representar las acciones que realiza cada uno de ellos dentro del sistema. Además, se muestra la relación de las historias de usuarios o casos de uso (“Registrar Paciente”, “Evaluar Paciente”, “Consultar Registro Paciente”, “Realizar Practica o Ejercicio”) con otras funciones que cumple el sistema.

Para explicar de forma breve como se relaciona el sistema con los diferentes actores y casos de uso involucrados, se toma a forma de ejemplo el caso de uso: “Consultar Registro Paciente” esta es una función que la puede realizar tanto el paciente como el administrador, el Actor administrador es el profesional encargado de supervisar el progreso del paciente. Además, el comportamiento de “Consultar Registro Paciente” es diferente o muestra diferentes

descripciones de escenarios como: “Resultado Datos Paciente”, “Resultados Evaluación Paciente” y “Resultados Desempeño Paciente” ya que están asociadas al mismo caso de uso por medio de la relación <<Extend>>. Por otra parte, estas funciones hacen algo en común y es: “Comprobar Datos en BD” ya que posee una relación de tipo <<Include>> o <<uses>>.

Así mismo, se diseñó el diagrama de clases con el fin de visualizar las relaciones entre las clases que involucran el sistema. Como se puede observar en la Figura 4.3.

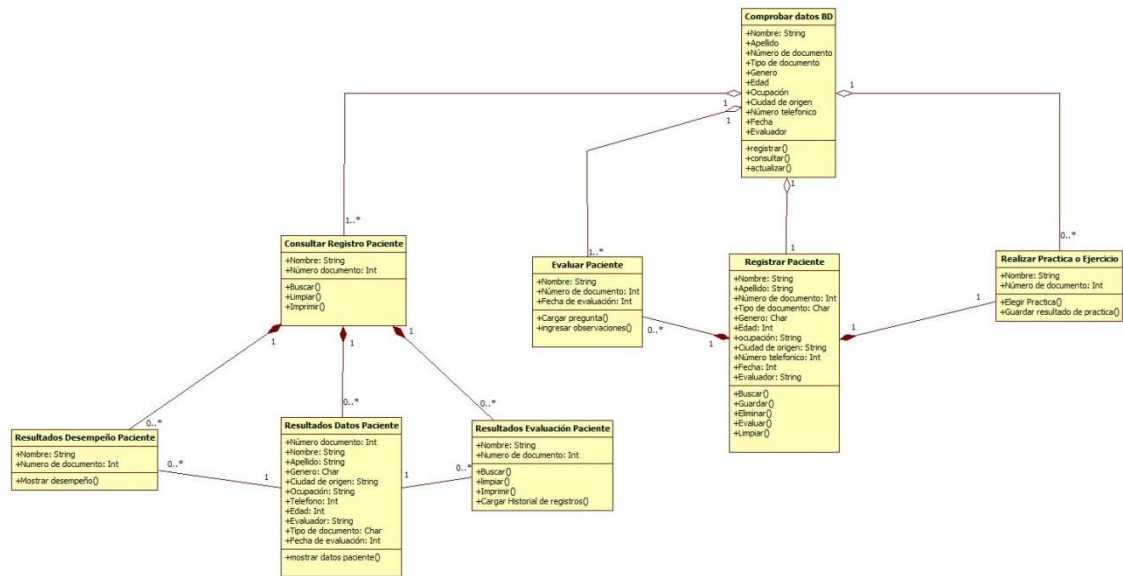


Figura 4.3. Diagrama de Clases del sistema. Fuente propia

Teniendo en cuenta, la Figura 4.2 existe un caso de uso “realizar practica o ejercicio” con este se hace referencia al ambiente virtual del sistema. Para lo cual, se utiliza un modelado de grafos para la estructura de tipo árbol que describe todos los componentes del ambiente de JAVA 3D. Para construir un escenario gráfico en JAVA 3D [27] es necesario partir del hecho que existe un universo virtual que puede contener todas las escenas gráficas del ambiente virtual. En nuestro caso consideramos un escenario gráfico para implementar el ambiente, este está representando un conjunto de objetos que definen la geometría, luces, localización, orientación y la apariencia de los objetos que ahí se visualizan. Así como se muestra en la Figura 4.4.

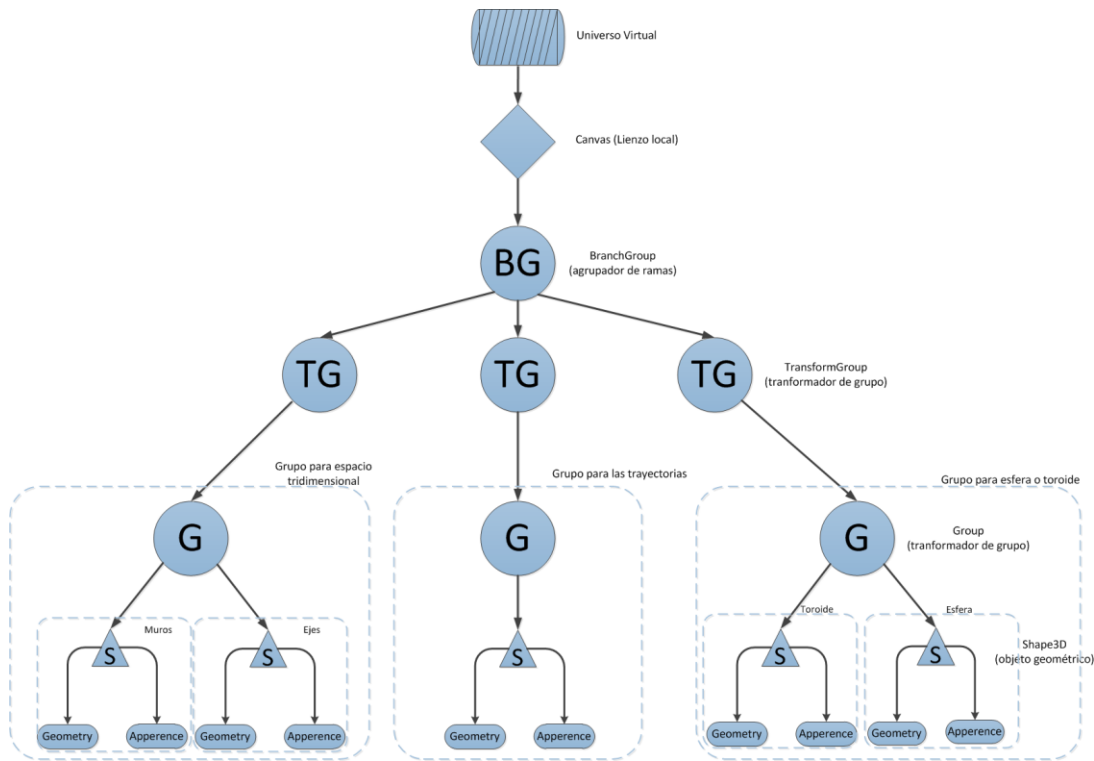


Figura 4.4. Grafo del árbol de JAVA 3D. Fuente propia

1.1.1.3. Fase de desarrollo y pruebas

Plan de entregas y pruebas

Para la interfaz gráfica de usuario del prototipo desarrollado en este trabajo de grado se plantearon 3 historias de usuario, descritas en el anterior ítem, de las cuales cada una tiene su respectivo plan de entregas y pruebas. Como se define a continuación:

1. Entrega 1 (Interfaz gráfica de usuario):

Esta entrega contiene la visualización de la interfaz gráfica de usuario del sistema. Consiste en visualizar una interfaz amigable y entendible. Además, se puede ver los botones de las funciones Registro Pacientes, Consulta Pacientes, Práctica y Ejercicios.

1.1. Prueba 1 (Interfaz gráfica de usuario):

A continuación en Figura 4.5, se muestra la interfaz gráfica del sistema:

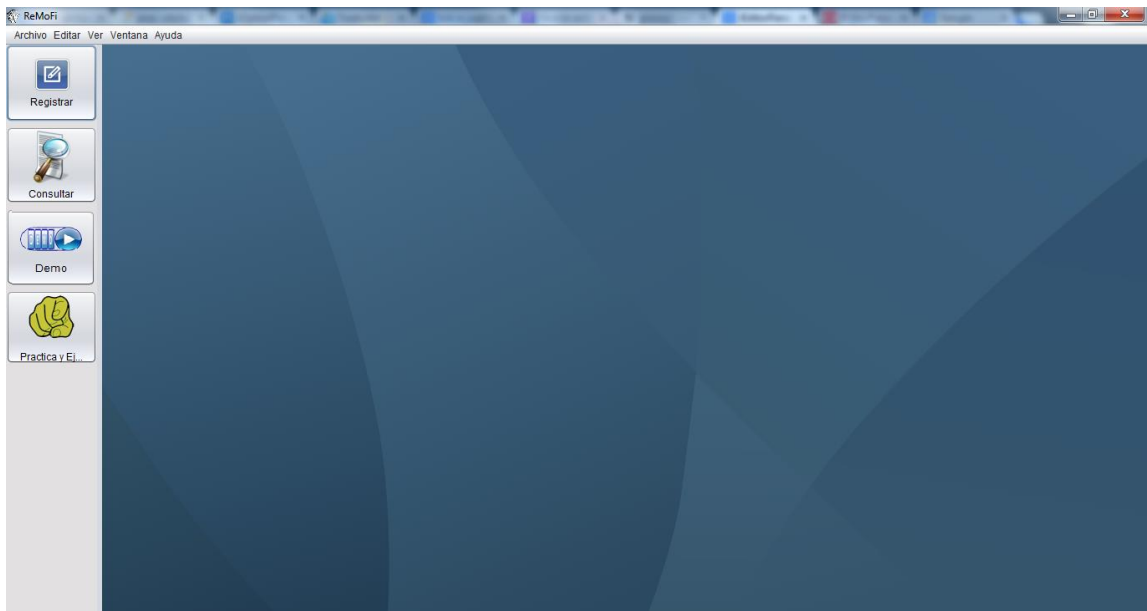


Figura 4.5. Interfaz gráfica de usuario. Fuente propia

La anterior figura, muestra las opciones descritas en la historia de usuario general. Donde están las funciones: Registrar, Consultar, Práctica y Ejercicios. En este caso dichas funciones son representadas en la interfaz como botones dentro de la ventana de la interfaz gráfica.

2. Entrega 2 (Registro Pacientes):

Registra los datos personales del paciente. Además, están habilitadas las funciones de: buscar, guardar, eliminar y limpiar en la interfaz gráfica. Cabe aclarar, la opción limpiar cumple con la función de borrar la información escrita en las casillas del formulario "Registro Pacientes" esto con el fin de tener el formulario limpio para realizar un nuevo registro.

2.1. Prueba 2 (Registro Pacientes):

A continuación, en la Figura 4.6 se muestra la interfaz gráfica de la función Registro Pacientes. Se puede observar el formulario 1 "REGISTRAR" que consiste en el registro de los datos personales del paciente con sus respectivos campos de texto asignados en la interfaz, esta información es almacenada en la base de datos del sistema. Además, tiene las funciones de buscar, guardar, eliminar, limpiar y evaluar que se encuentran en la parte inferior de la interfaz y están representados por medio de botones.

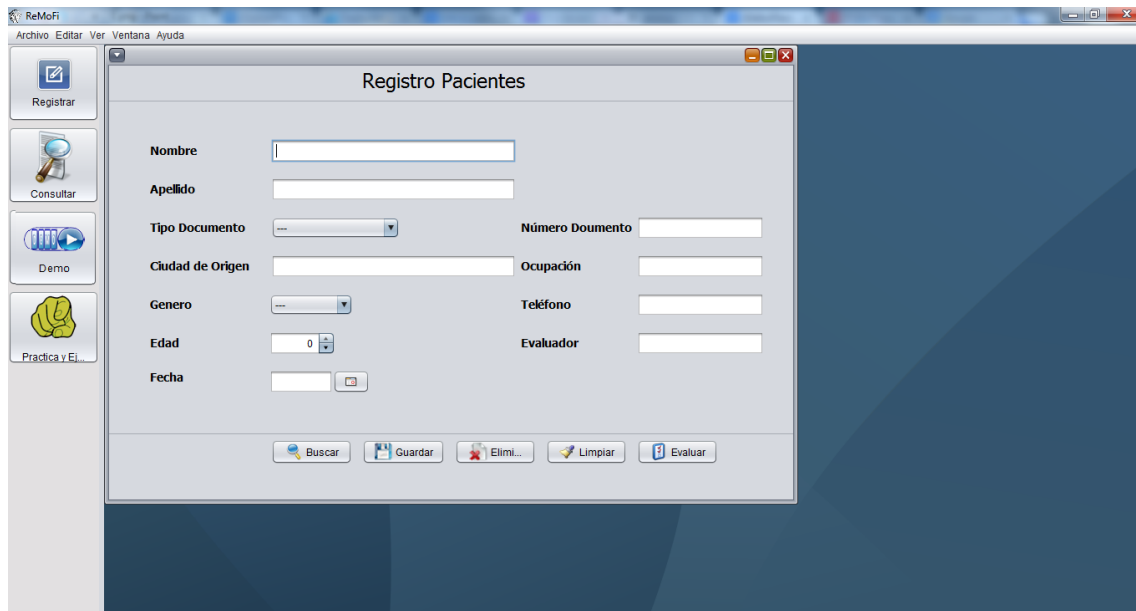


Figura 4.6. Interfaz registro pacientes. Fuente propia

Cabe aclarar, dentro de la base de datos del sistema ya se encuentran registrados algunos pacientes, esto con el fin de probar si efectivamente la base de datos almacena la información que se desea guardar en cada formulario.

Teniendo en cuenta lo anterior, para realizar la prueba de registro se pretende registrar un número de documento ya existente en la base de datos, el número de documento prueba es 123. A continuación, en la Figura 4.7 se muestra la prueba antes mencionada a la función Registrar Pacientes.

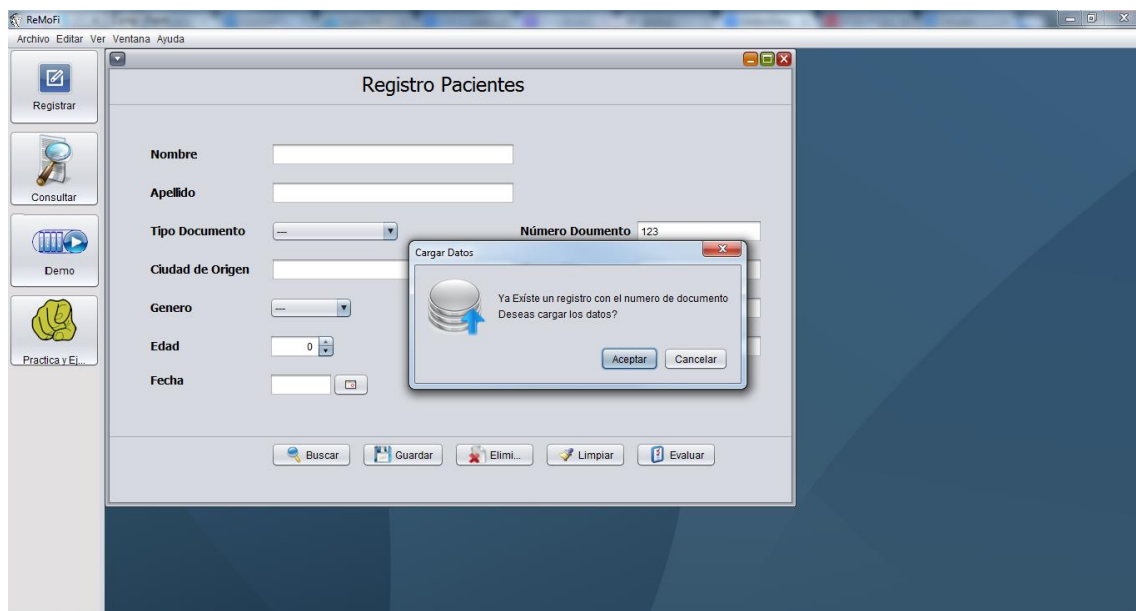


Figura 4.7. Interfaz registro paciente ingresando número de documento. Fuente propia

De acuerdo a lo anterior, se puede verificar que la base de datos funciona correctamente en la función “registro pacientes”. Ya que, al ingresar el número de documento 123 el sistema rebota un mensaje donde expresa que dicho registro con el número de documento mencionado ya se encuentra registrado. Posteriormente, pregunta si desea cargar los datos que se encuentran relacionados con dicho documento. Finalmente, se da clic en el botón aceptar para que enseguida cargue el registro relacionado a ese número de documento ingresado. Así como, se muestra en la Figura 4.8

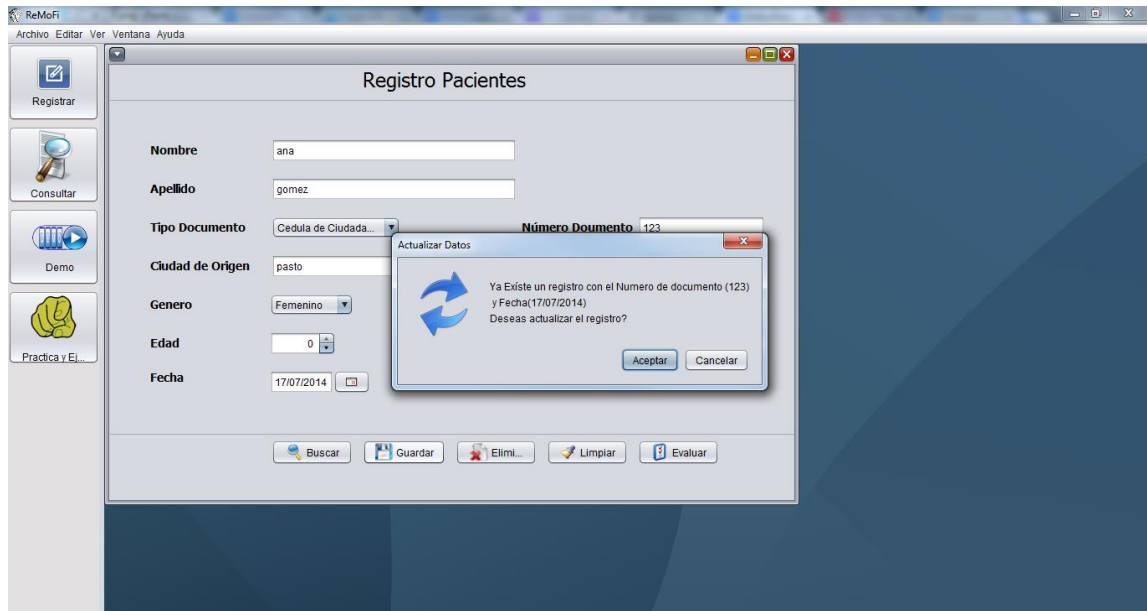


Figura 4.8. Registro de pacientes almacenado previamente en la base de datos. Fuente propia

Teniendo en cuenta Figura 4.8, se puede observar que la base de datos verifica el número de documento ingresado junto con la fecha de registro realizada para dicho documento dentro de los datos guardados en la base de datos. Además, en el mensaje de confirmación que rebota, el sistema pregunta si desea actualizar el registro el cual posee dos opciones “aceptar” o “cancelar”, dichas opciones están en forma de botones para ser mostradas en la interfaz gráfica.

3. Entrega 3 (Evaluación paciente):

Por medio de la información suministrada por parte del área de fisioterapia de la universidad del Cauca se obtuvieron las características de la evaluación que se realiza al paciente, donde se especifica por medio de un cuestionario las diferentes habilidades que puede presentar un paciente y de acuerdo a estas preguntas el especialista encargado establece en qué condiciones se encuentra el paciente para realizar posteriormente los ejercicios de rehabilitación.

Dentro del formulario “REGISTRAR”, existe una opción en la parte inferior de la interfaz con el nombre de evaluar, ahí el paciente puede responder a las preguntas relacionadas con motricidad fina de miembro superior.

3.1. Prueba 3 (Evaluación pacientes):

Una vez registrado el paciente, se procede a realizar la evaluación que contiene una serie de preguntas relacionadas con habilidades del miembro superior. La interfaz gráfica muestra el nombre del paciente con su respectiva cédula de ciudadanía, fecha de evaluación y responder pregunta. Así como, los botones anterior, siguiente e ingresar observaciones. A continuación se presenta la Figura 4.9

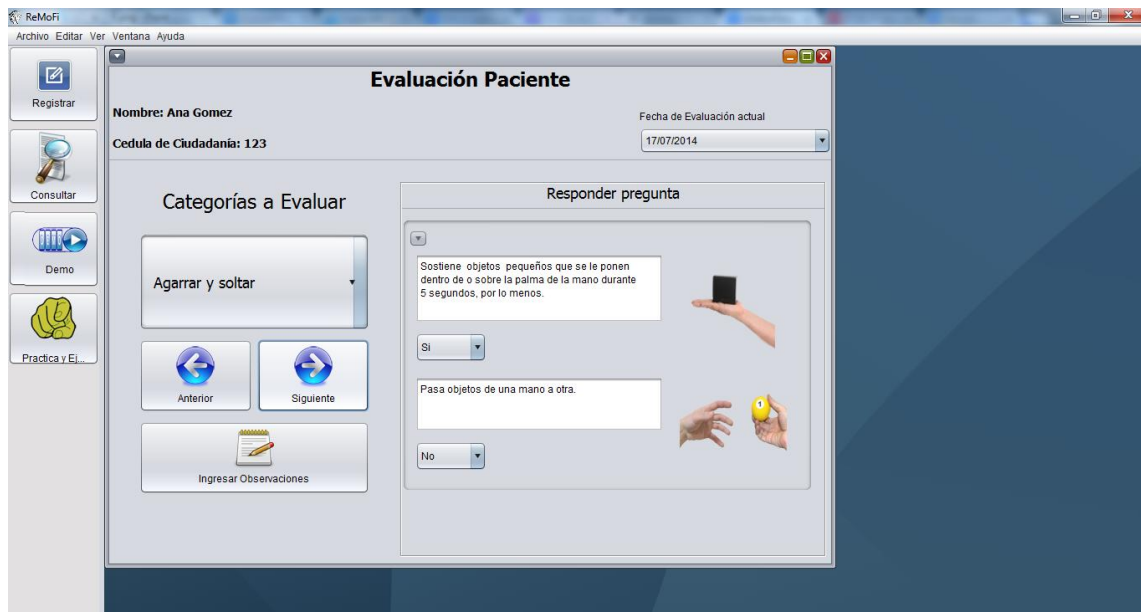


Figura 4.9. Interfaz evaluación paciente. Fuente propia

Cuando el paciente finaliza la evaluación el profesional a cargo puede ingresar las observaciones que tiene respecto a la evaluación del paciente. Así como, se puede observar a continuación en la Figura 4.10

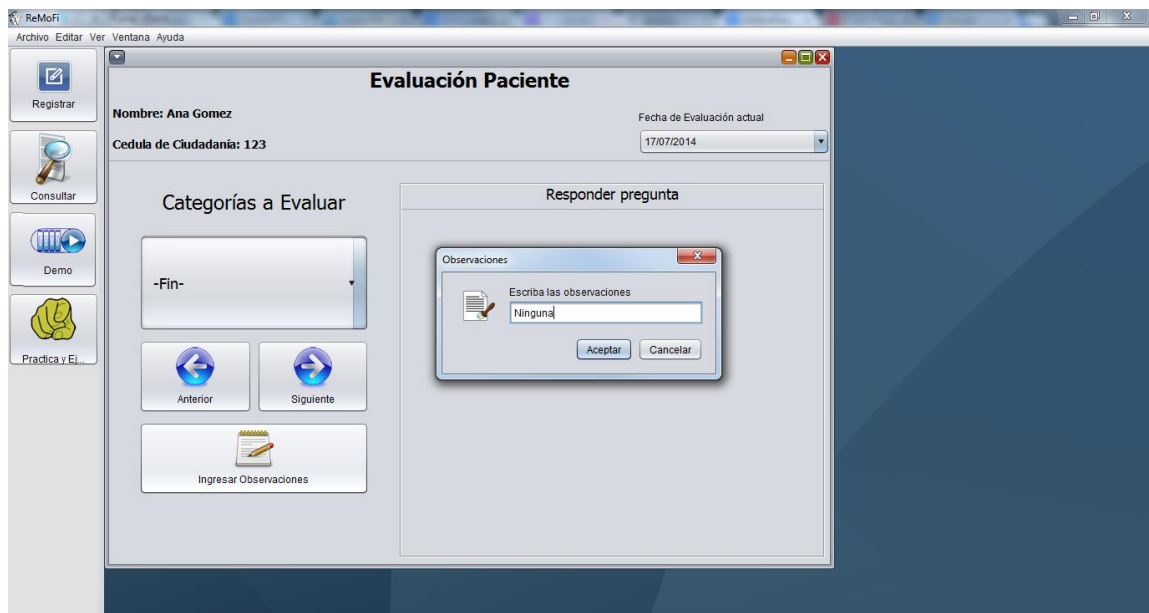


Figura 4.10. Interfaz observaciones. Fuente propia

4. Entrega 3 (Consultar pacientes):

Esta interfaz permite visualizar el formulario 2 “CONSULTAR”. Para realizar la consulta de un paciente se debe ingresar el nombre o número de documento posteriormente se da clic en buscar para obtener la búsqueda deseada y el botón limpiar deja despejados los cuadros de texto para iniciar una nueva consulta. Además, la interfaz presenta en la parte inferior de la ventana unas pestañas llamadas: “Resultados datos pacientes”, “Resultados evaluación paciente”, “Resultado desempeño pacientes” y “Resultado desempeño pacientes sigma”.

4.1. Prueba 3 (Consultar pacientes):

En la Figura 4.11, se muestra la interfaz gráfica de la función consultar datos pacientes. Donde se puede realizar la consulta ingresando el nombre o número de documento.

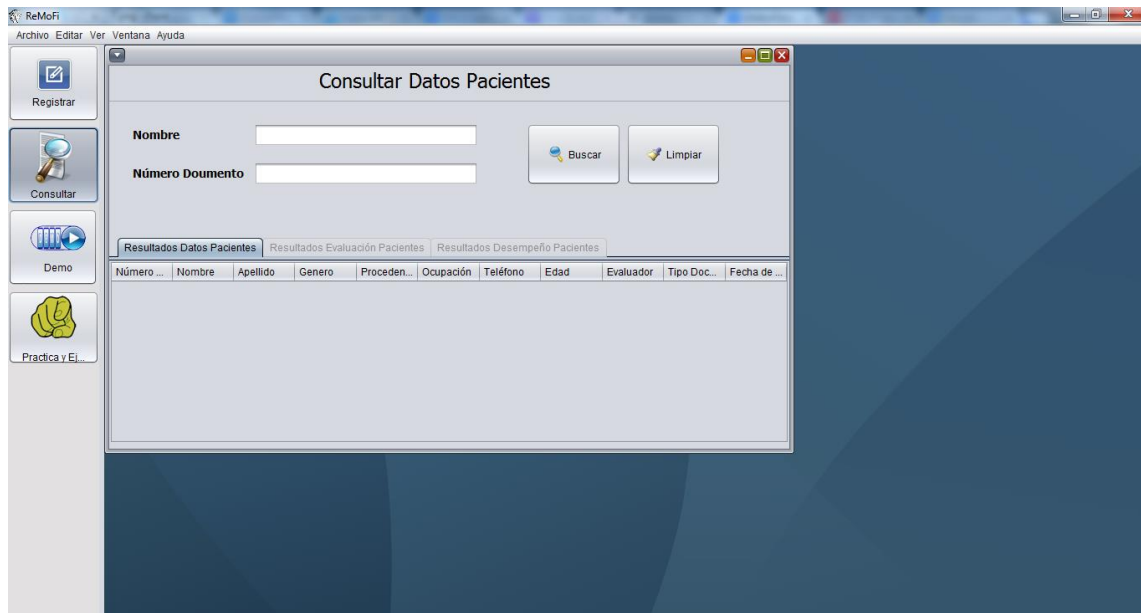


Figura 4.11. Interfaz consultar datos pacientes. Fuente propia

Para realizar las pruebas pertinentes se desea consultar un paciente ingresando la inicial de su nombre, en este caso se decide hacer la consulta con la letra “e”, Así como se muestra en la Figura 4.12

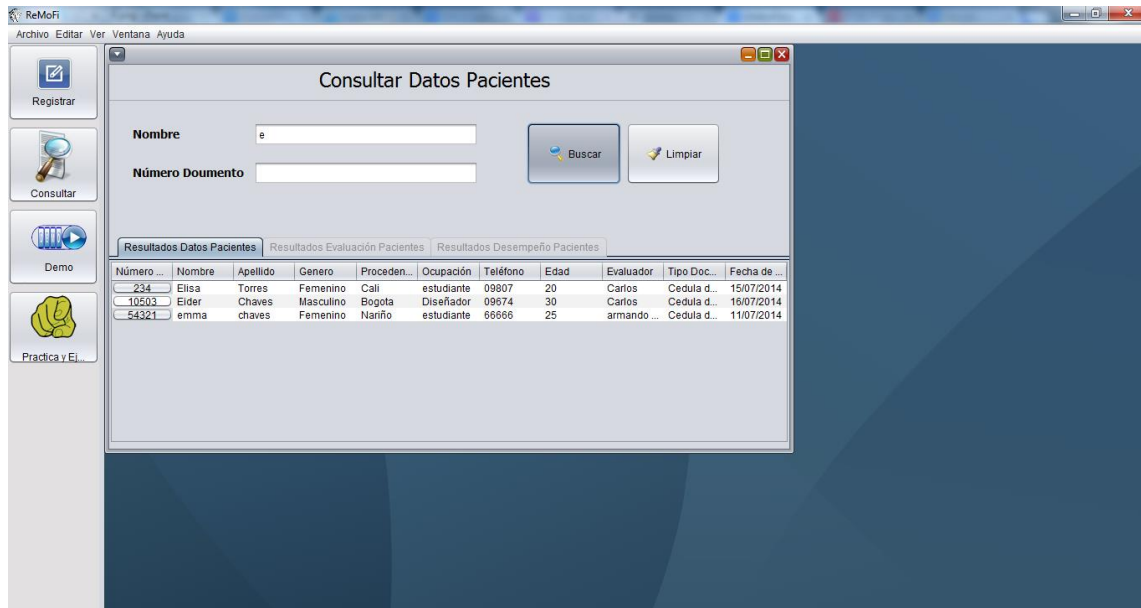


Figura 4.12. Interfaz consulta datos pacientes probando el campo de texto "Nombre". Fuente propia

Así como se puede observar en la Figura 4.12, el sistema realiza la búsqueda en la base de datos con todos los pacientes registrados que coincidan con el nombre que empieza por la letra “e”. Posteriormente, el sistema arroja una lista

con todas las coincidencias, en este caso muestra 3 pacientes con sus respectivos datos personales que coinciden con la consulta esperada.

Además, si se desea una búsqueda más precisa existe la opción de consulta por medio del número de documento como se muestra a continuación en la Figura 4.13

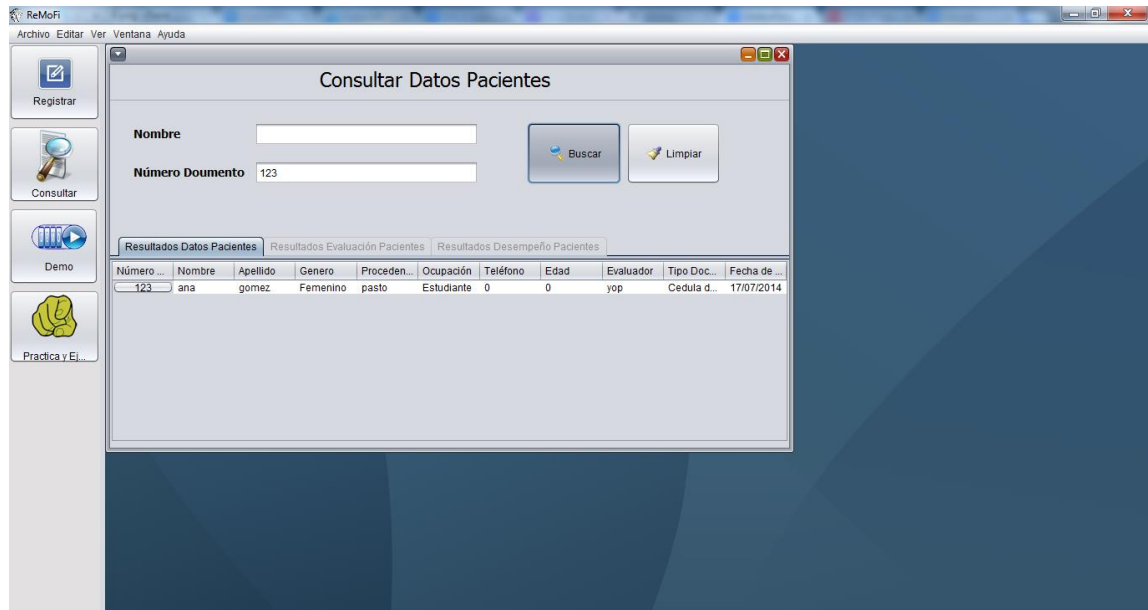


Figura 4.13. Interfaz consulta datos pacientes probando el campo de texto “Número Documento”.
Fuente propia

Como se puede observar en la Figura 4.13, se toma como prueba el número de documento “123” para llevar a cabo la consulta dentro de la base de datos. El sistema realiza la búsqueda dentro de los registros guardados, posteriormente es encontrado y finalmente es mostrado en la interfaz gráfica con el nombre de: Ana Gómez.

Por otro lado, existe la función de ver los resultados de evaluación de los pacientes. Este muestra la evaluación del paciente con sus respectivas respuestas. Para esta prueba se toma un usuario de la Figura 4.13, con el nombre de “Ana Gómez” y el sistema muestra la evaluación que respondió dicho paciente. Además, el sistema muestra el número de registros que realizó el usuario con sus respectivas fechas y un botón con la opción de imprimir. Así como se muestra a continuación en la Figura 4.14

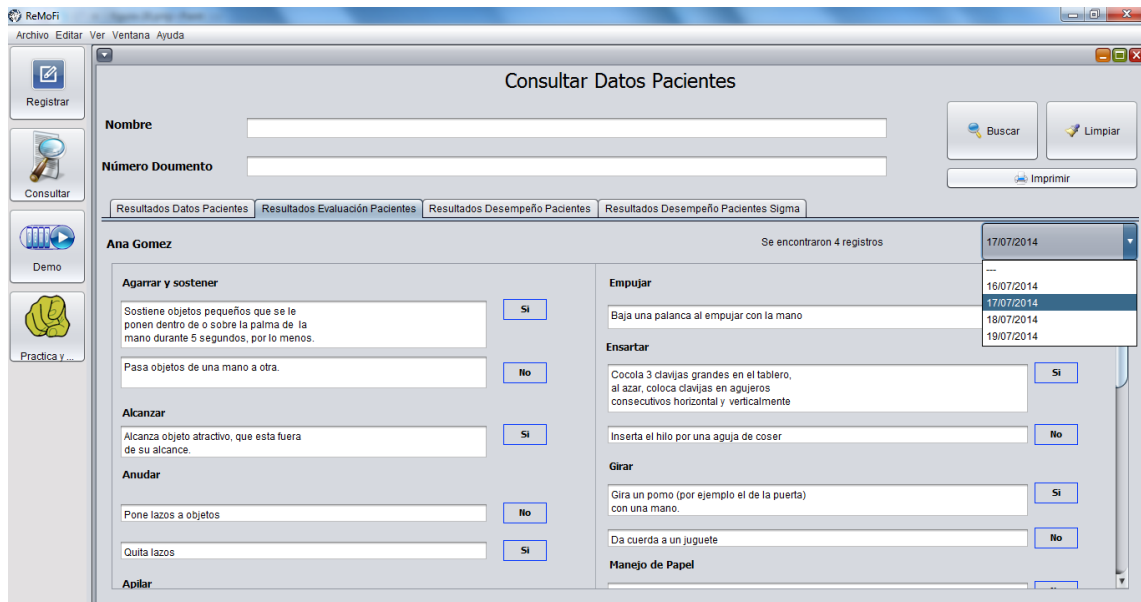


Figura 4.14. Interfaz gráfica de resultados evaluación pacientes. Fuente propia

Cuando se da clic en el botón “imprimir”, este despliega las opciones de impresión para imprimir la evaluación del paciente, así como se muestra en la Figura 4.15.

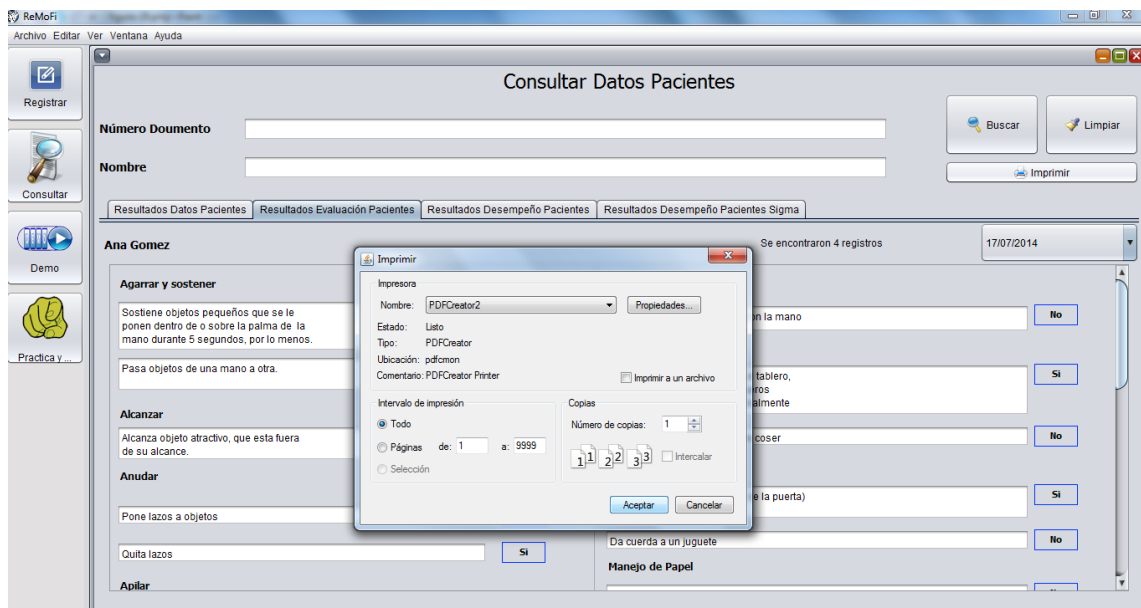


Figura 4.15. Interfaz opción imprimir evaluación. Fuente propia

Además, el profesional en fisioterapia puede observar el progreso que tiene el paciente por medio de las gráficas que arroja el ambiente virtual en la opción “Resultado Desempeño Pacientes”. Este progreso es visible para cada práctica realizada por parte del usuario. Así como, se muestra en la Figura 4.16. La

curva muestra las practicas hechas en cada nivel teniendo en cuenta un score en un rango de 0 a 100; tomando a 0 como el puntaje más deficiente y 100 el máximo. Es decir, que esta curva determina el número de errores cometidos durante la realización del ejercicio, tomando como error las veces que el usuario se sale de la trayectoria cada vez que este intenta recorrerla.

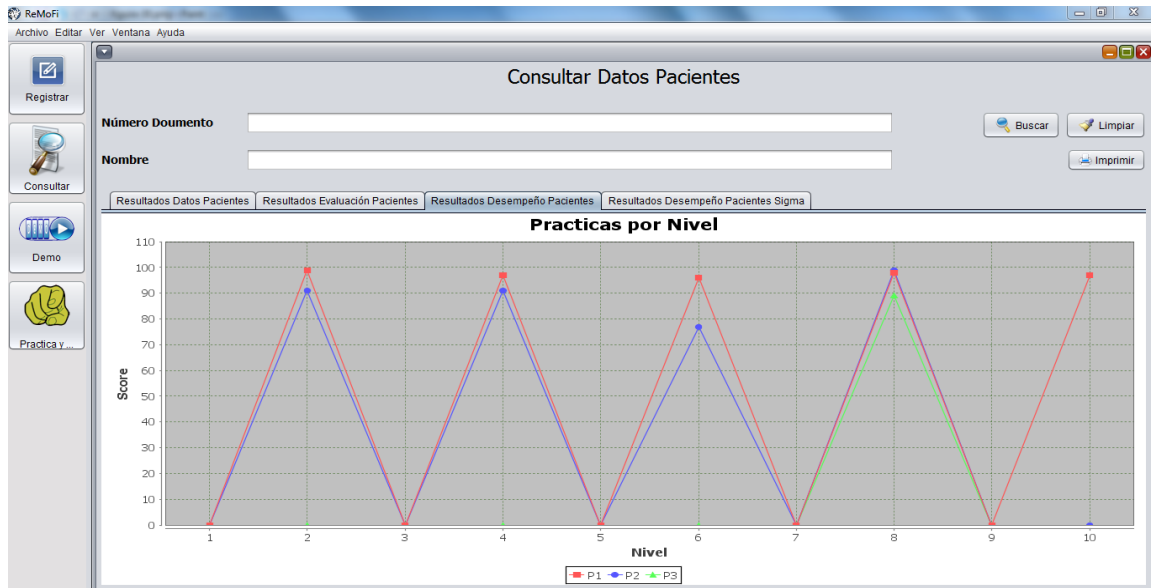


Figura 4.16. Resultado desempeño paciente. Fuente propia

De igual manera, el sistema grafica la desviación estándar para cada nivel teniendo en cuenta el número de prácticas que el paciente realiza. Tal como, se muestra en la siguiente Figura 4.17; en el capítulo 5 se explica con más detalle la manera en la que se determina la desviación estándar en el ambiente virtual.

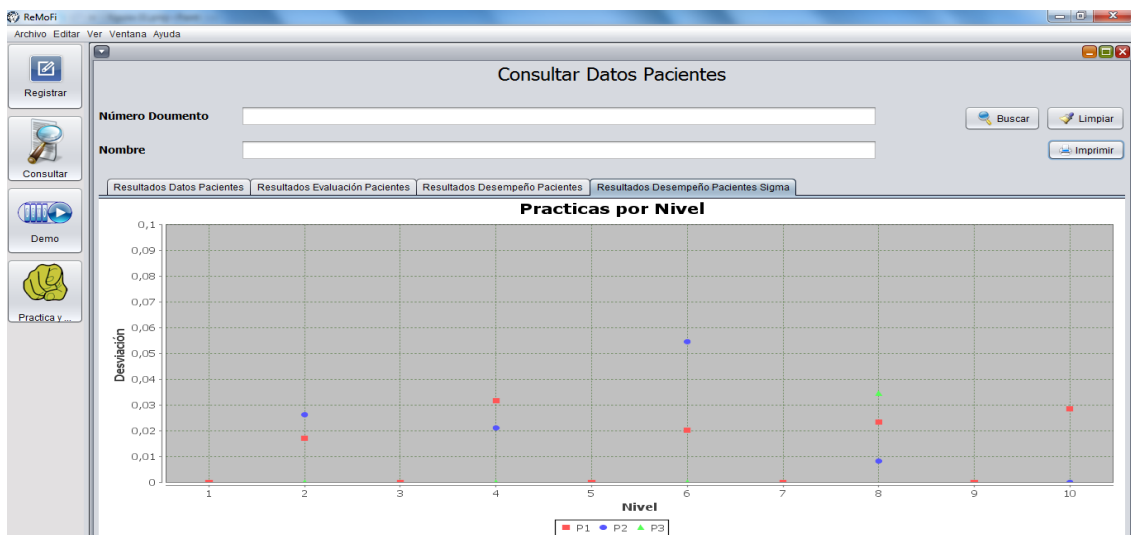


Figura 4.17. Resultado Desempeño Pacientes sigma (Desviación Estándar). Fuente propia

4.2. Descripción ambiente virtual

Dentro de la aplicación se implementan dos opciones para el ambiente virtual; la primera es un acceso a un DEMO que permite al usuario entender cómo funcionan los ejercicios de entrenamiento y la segunda es el ambiente virtual donde se cargan los ejercicios de entrenamiento, cada uno con una trayectoria diferente.

4.2.1. DEMO

Esta opción brinda al usuario el acercamiento al sistema y por medio de una animación se explica la forma en la cual se debe utilizar el dispositivo. En el anexo A se encuentra detallado la instalación y configuración del dispositivo *LeapMotion*. A continuación, se muestra en la Figura 4.18 la opción DEMO en el ambiente virtual.

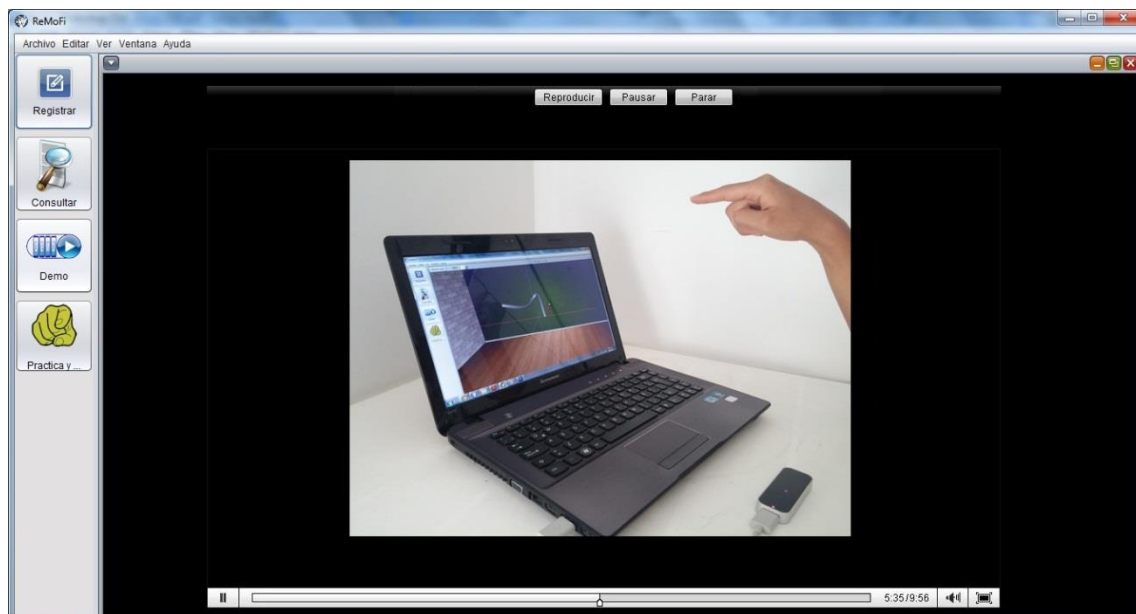


Figura 4.18. Opción DEMO en el ambiente virtual. Fuente propia

4.2.2. Práctica de los ejercicios de entrenamiento

El botón “*prácticas*” inicia la sesión de usuario para realizar el ejercicio de rehabilitación y muestra las trayectorias generadas aleatoriamente para los ejercicios de entrenamiento de rehabilitación de motricidad fina de miembro superior. Para ingresar a los ejercicios es necesario haber iniciado una sesión; por lo tanto, el sistema arroja una ventana de dialogo donde solicita ingresar el número de identificación del paciente. Así como, se muestra en la Figura 4.19

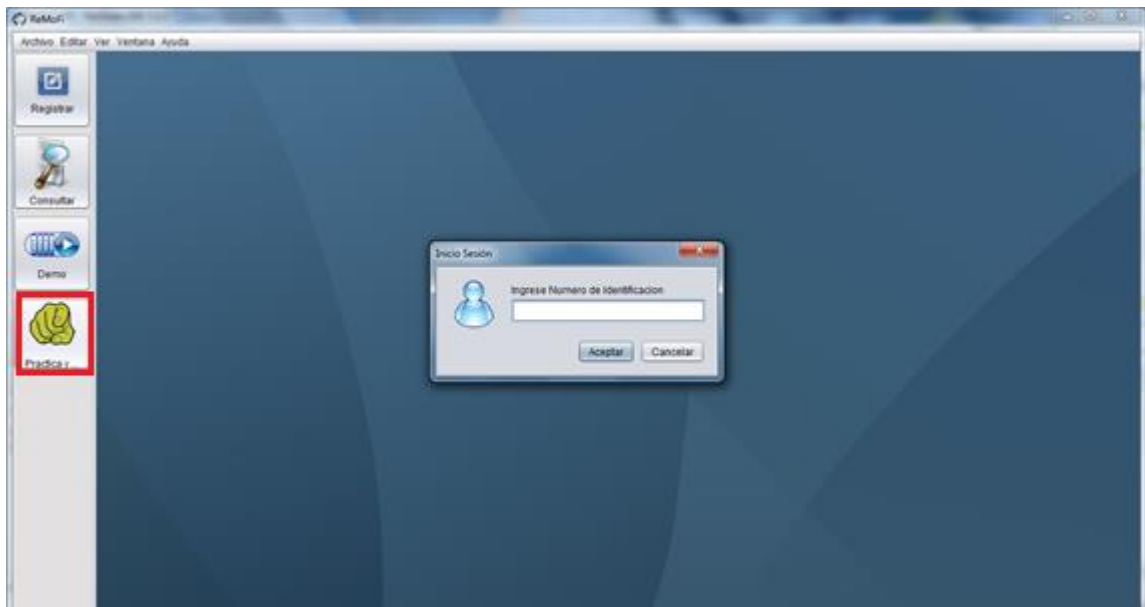


Figura 4.19. Opción "practicass" en el ambiente virtual. Fuente propia

Cabe aclarar, existe otra manera de iniciar sesión para las prácticas del ambiente virtual; por medio de las opciones desplegadas en el botón archivo de la ventana principal de la interfaz gráfica. Como se observa en la Figura 4.20

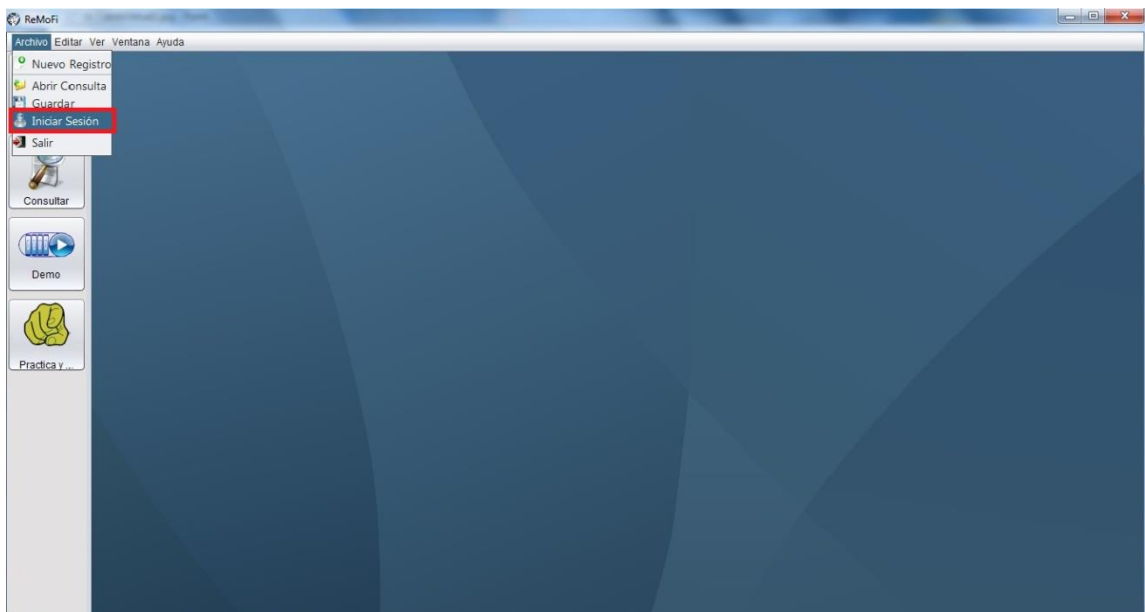


Figura 4.20. "Iniciar Sesión" para la práctica de ejercicios en el ambiente virtual. Fuente propia

Adicionalmente, si el número de identificación que ingresa el paciente al sistema no se encuentra, este muestra un mensaje advirtiéndole que el registro no fue encontrado; así como, se muestra en la Figura 4.21 y el sistema vuelve a mostrar la ventana de diálogo de la Figura 4.19

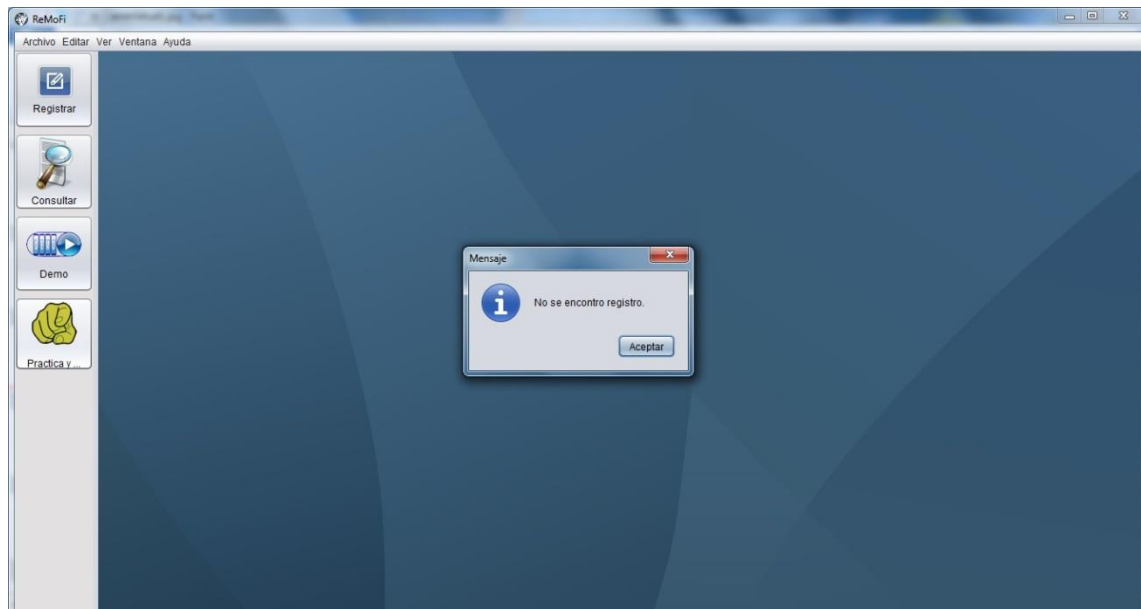


Figura 4.21. Ventana de diálogo advirtiendo que el usuario no se encuentra registrado. Fuente propia

Una vez que el usuario pueda iniciar sesión el sistema lo lleva al ambiente virtual. En este se puede visualizar un ambiente tridimensional de una habitación, la trayectoria generada aleatoriamente por medio del botón “*nueva curva*”, los niveles de complejidad que el usuario escoja, el tiempo que tarda el paciente en realizar el ejercicio y la opción Activar toro, que indica que el cursor esfera puede cambiar a una anillo. Así como, el nombre y número de identificación del paciente. Tal como, se muestra en la Figura 4.22

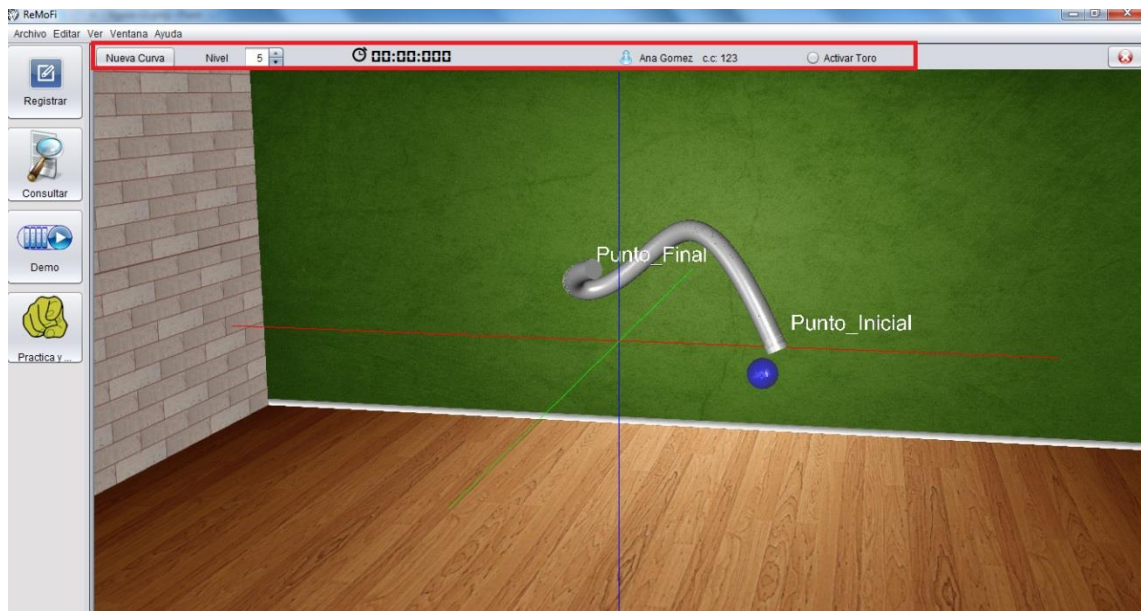


Figura 4.22. Inicio de sesión para un usuario registrado en el sistema. Fuente propia

Las trayectorias generadas aleatoriamente por el sistema tienen niveles de dificultad, estos están en un rango de 1 a 10; siendo 1 el nivel más fácil y 10 el nivel con mayor complejidad. Además, cada nivel posee N trayectorias que pueden ser realizadas por el usuario, esto asegura que no se repita una trayectoria. Cabe aclarar, la esfera o toroide (anillo) que sirven como cursor guía para viajar a través de la trayectoria cambia de color de acuerdo a su ubicación en la misma; por lo tanto, si el cursor está en color azul indica que es el inicio de la práctica y que aún el usuario no ha tocado la curva, en color rojo significa que no está realizando de forma correcta el ejercicio y en color verde que está recorriendo adecuadamente la trayectoria. A continuación, se puede observar las diferentes trayectorias generadas en el ambiente virtual.

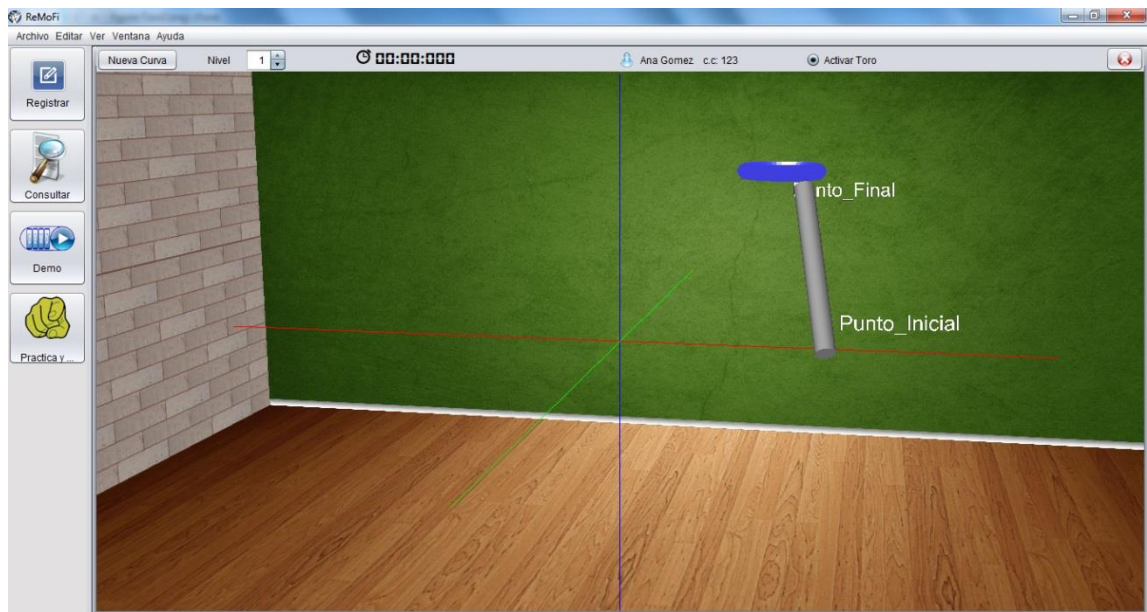


Figura 4.23. Trayectoria generada aleatoriamente, en el nivel de complejidad 1, teniendo activo cursor toroide. Fuente propia

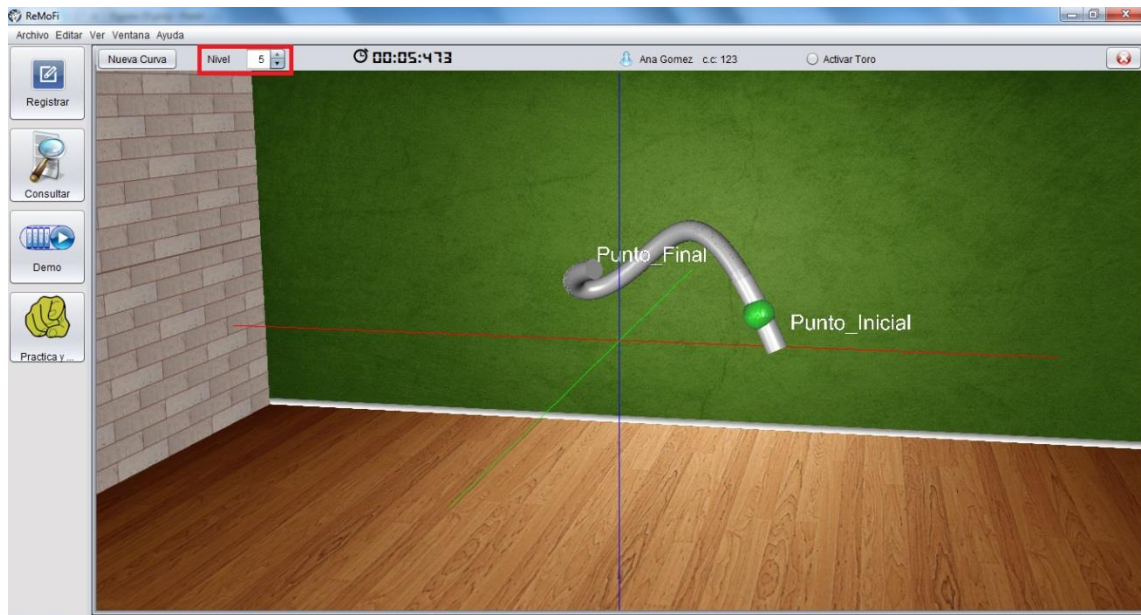


Figura 4.24. Trayectoria generada aleatoriamente, la esfera verde indica que se encuentra recorriendo la curva. Fuente propia

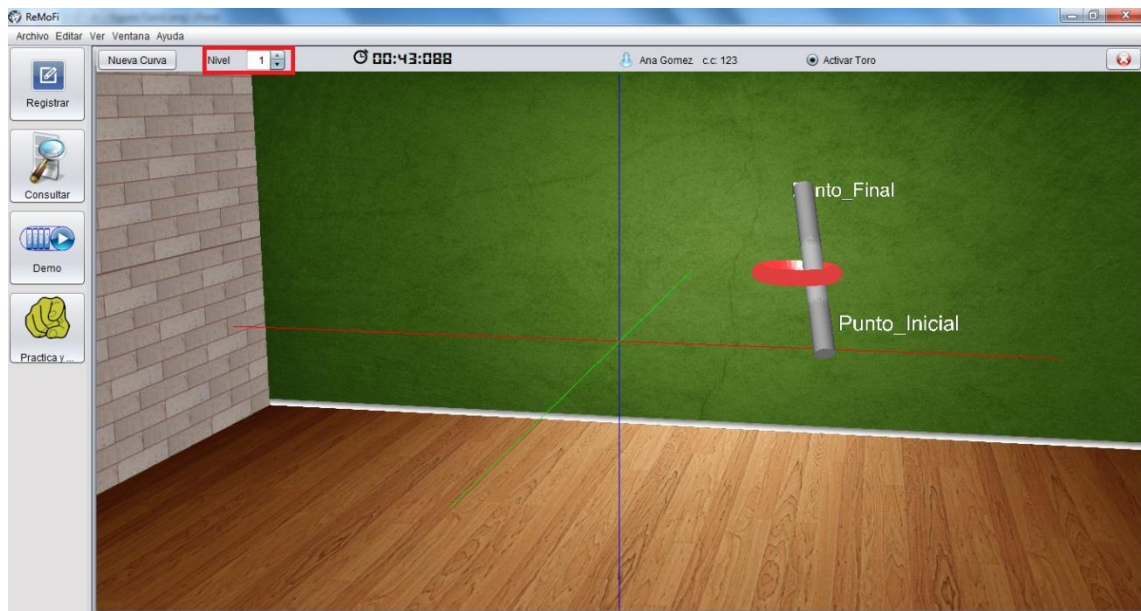


Figura 4.25. Trayectoria generada aleatoriamente, la esfera roja indica que se encuentra por fuera de la curva. Fuente propia

A continuación, se puede observar las diferentes trayectorias generadas en el ambiente virtual, para realizar la demostración se tomó los niveles 1, 3, 8 y 10.

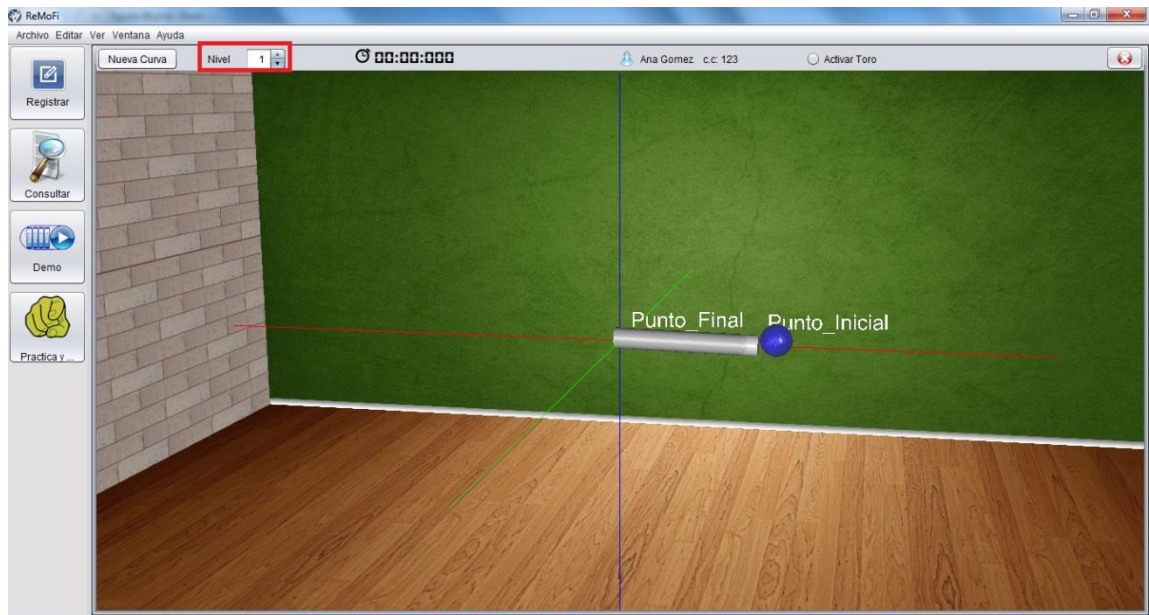


Figura 4.26. Trayectoria generada aleatoriamente con nivel de complejidad 1. Fuente propia

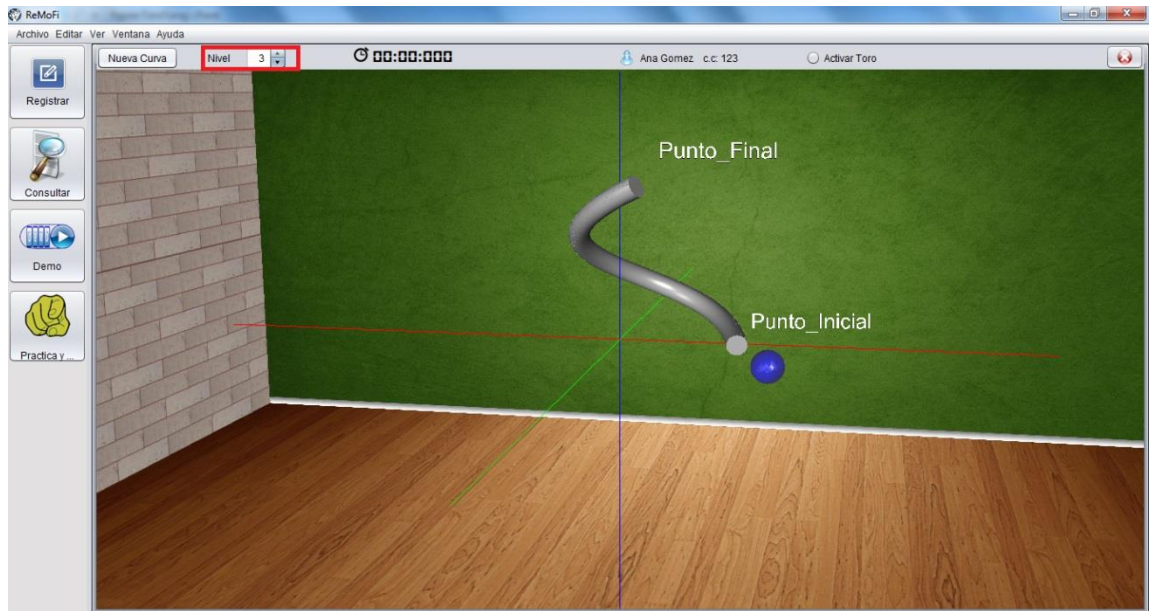


Figura 4.27. Trayectoria generada aleatoriamente con nivel de complejidad 3. Fuente propia

En las figuras siguientes, se muestra la trayectoria para el nivel 8 generada de forma aleatoria.

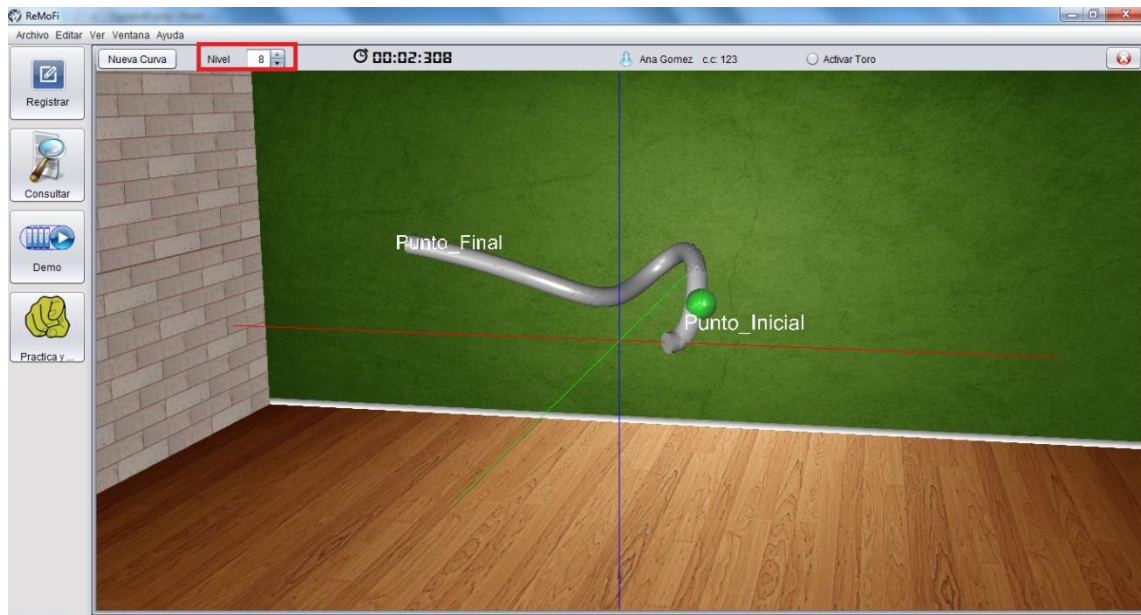


Figura 4.28. Trayectoria generada aleatoriamente con nivel de complejidad 8. Fuente propia

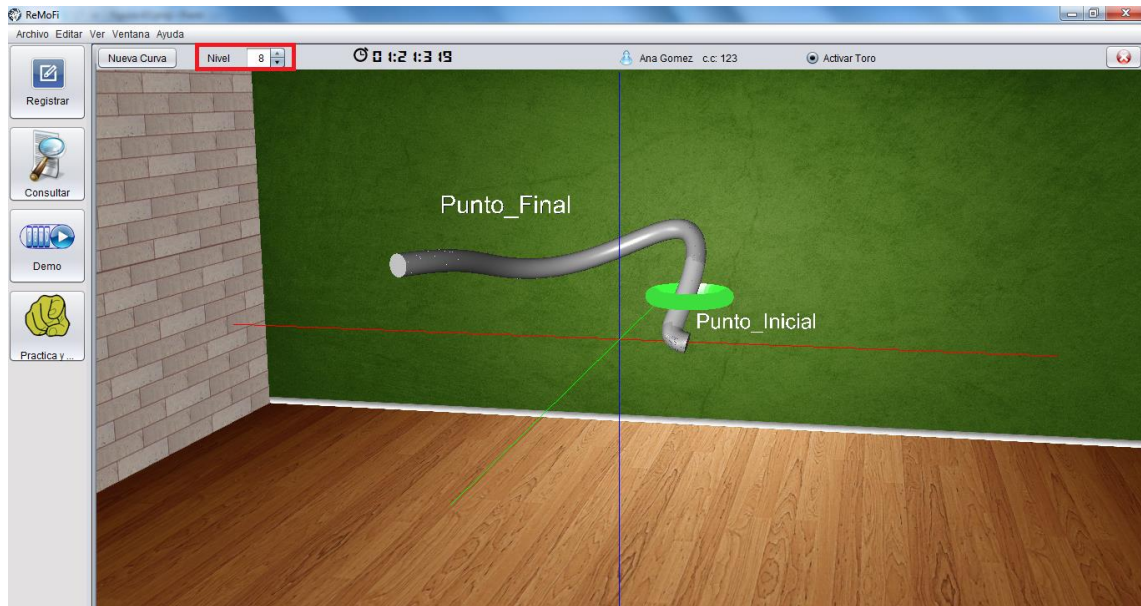


Figura 4.29. Trayectoria generada aleatoriamente con nivel de complejidad 8, toroide. Fuente propia

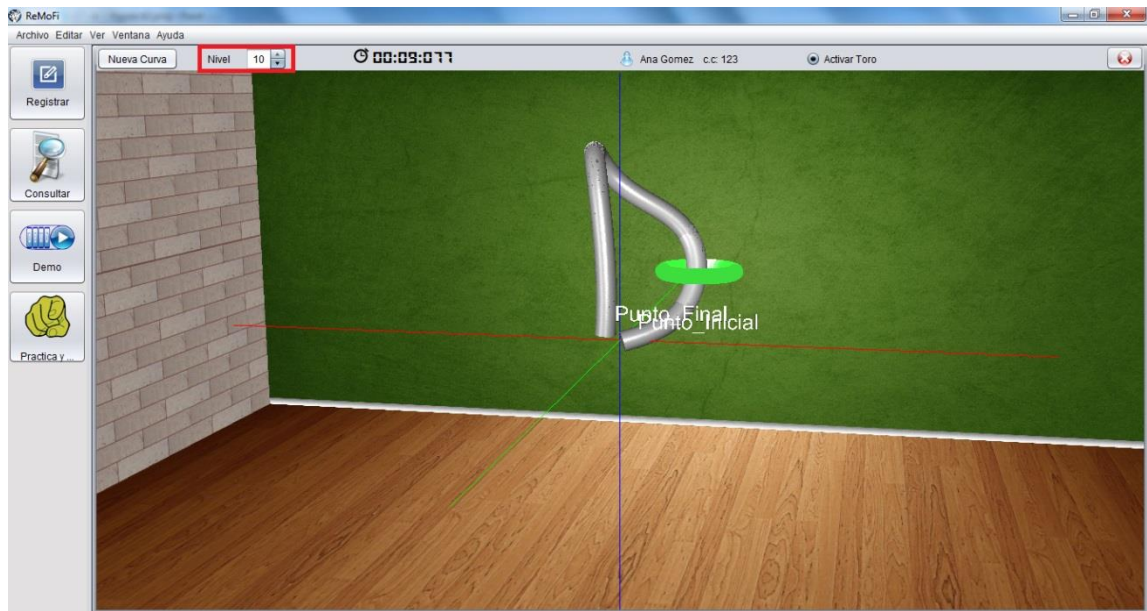


Figura 4.30. Trayectoria generada aleatoriamente con nivel de complejidad 10. Fuente propia

Finalmente, cuando el usuario haya terminado su práctica de ejercicios puede terminar su sesión por medio de la opción "Cerrar Sesión" esta se encuentra en las opciones desplegadas en el botón archivo de la ventana principal de la interfaz gráfica. Como se observa en la Figura 4.31



Figura 4.31. "Cerrar Sesión" para la práctica de ejercicios en el ambiente virtual. Fuente propia

CAPÍTULO 5

PRUEBAS DE EVALUACIÓN Y ANALISIS DEL AMBIENTE VIRTUAL

En este capítulo se describen las pruebas y el análisis realizado al ambiente virtual con realimentación visual por medio del dispositivo *LeapMotion*. Teniendo en cuenta lo anterior, se toma un grupo de personas con edades entre los 18 y 28 años para realizar dichas pruebas. Con el fin de analizar, las siguientes variables: tiempo en que el participante tarda en realizar el ejercicio y desviación estándar. Estos datos se determinan a partir de las pruebas realizadas a cinco personas durante diez días utilizando tres de los diez niveles de complejidad que posee el ambiente virtual. Finalmente, se analizan los resultados para establecer la evolución del desempeño del grupo de personas antes mencionado. Con esto se busca determinar la eficiencia del ambiente virtual, teniendo en cuenta lo siguiente: primero es establecer que el usuario pueda seguir las trayectorias generadas por el sistema en el ambiente virtual; segundo, verificar que el dispositivo incorporado al ambiente virtual sirve como medio para la ejecución de los entrenamientos o ejercicios y finalmente hacer seguimiento de la ejecución de las trayectorias en las diferentes sesiones de prueba.

5.1. Consideraciones para la prueba de evaluación del ambiente virtual

El presente trabajo de grado está enfocado en el desarrollo e implementación de un ambiente virtual cuyo propósito es el de servir a futuro como un medio para rehabilitación fina de miembro superior. Cabe aclarar, que no es un trabajo cuyo fin sea realizar un estudio clínico o investigativo sobre rehabilitación. Teniendo en cuenta lo anterior y con el fin de realizar las pruebas al ambiente virtual, se consideró un grupo de personas con edades entre 18 y 28 años de la Ciudad de Popayán. Para la elección de los participantes a las pruebas se tuvo en cuenta que cumplieran con las siguientes características:

- No presentan problemas físicos notorios del miembro superior.

- No presentan problemas visuales, que les impida diferenciar los colores y las formas del ambiente virtual.
- Pueden realizar la mayoría de las habilidades motrices descritas en el cuestionario de evaluación de participante. Como, atarse los cordones de los zapatos, agarrar un objeto con la mano, etc.
- La persona debe tener la disponibilidad de tiempo para facilitar la realización de las prácticas durante diez días.

Por otro lado, se trabaja bajo la hipótesis de que con 40 cuadros por segundo se pueden muestrear fielmente los movimientos reales del humano que pueden ser registrados por el *LeapMotion* (con un máximo de 230 cuadros por segundo). Esto se cumple debido a que, la estructura del código de programación del sistema presenta retardos máximos de 25 milisegundos por ciclo, esto quiere decir que realiza el paso por una línea de código 40 veces en un segundo con el propósito de generar el renderizado gráfico. Cabe mencionar, que en el mercado existen otro tipos de sensores que capturan los movimientos, teniendo como referencia al dispositivo de Microsoft Kinect este trabaja con 30 cuadros por segundo para capturar las imágenes captadas por su cámara [31] y el PlayStation Move de SONY que reconoce de manera precisa la posición del objeto en el espacio para cumplir este objetivo este dispositivo necesita de un elemento externo a él como lo es la cámara PlayStation Eye, esta permite capturar un máximo de 60 imágenes por segundo [32].

5.2. Pruebas de evaluación del ambiente virtual

Considerando que la realización de las pruebas es la fase final del desarrollo del ambiente virtual, se escogieron tres de los diez niveles de complejidad. Para lo cual, se adecuo la interfaz gráfica con tres accesos a los niveles de prueba escogidos (1, 5 y 7). Se realizó una generación aleatoria de trayectorias para los niveles antes mencionados y se capturaron los datos que las generaban para determinar cuáles serían las trayectorias para realizar la fase de prueba. De las cuales, el nivel 1 presenta una línea recta, posee esta característica dado que este nivel es el de menor complejidad; el nivel 5 siendo un nivel intermedio muestra una trayectoria con curvas pronunciadas en el trayecto de la misma y el nivel 7 en el ambiente virtual representa un nivel avanzado de práctica, este tiene curvas mucho más pronunciadas que el nivel 5 y anteriores a él. Tal como, se muestra en la Figura 5.1

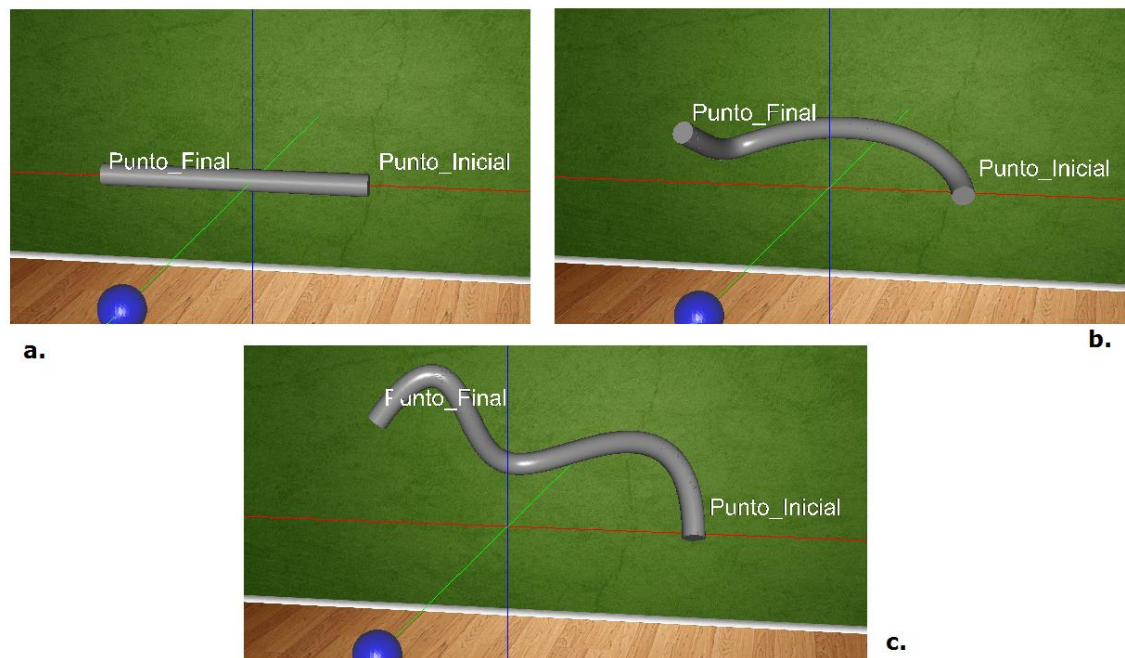


Figura 5.1. Trayectorias de prueba. a) Nivel 1 de complejidad, b) Nivel 5 de complejidad, c) Nivel 7 de complejidad. Fuente propia

En el momento de la realización de la prueba se presenta al participante el ambiente virtual y se explica en que consiste el ejercicio esto con el fin, de determinar que:

1. El participante debe comprender la equivalencia de sus movimientos con los generados en el ambiente virtual.
2. El participante tiene la oportunidad de escoger de forma libre con que cursor guía desea realizar las prácticas, ya sea con la esfera o el toroide.

Además, existe una persona que dirige la prueba que en adelante se llamará evaluador, es el encargado de guiar la práctica con el fin de verificar los datos obtenidos durante la realización de la misma. Por lo cual, se realizó unas pruebas de control que fueron obtenidas de forma manual por parte del evaluador, como:

1. Cronometrar la duración de cada ejercicio realizado por el participante.
2. Contar el número de errores cometidos por el participante durante la ejecución de la práctica. El número de errores son considerados como las veces que el participante sale de la trayectoria (con una distancia mayor al radio transversal de la trayectoria).
3. Verificar los datos antes mencionados y comparar los resultados que arroja el ambiente virtual.

5.3. Registro de los datos de la prueba del ambiente virtual

Como se mencionó anteriormente, se escoge para realizar la prueba del ambiente virtual 3 niveles de complejidad con una curva específica en cada uno (las mismas curvas para todos los participantes); esto se realiza con el propósito de poder comparar el progreso de los resultados de cada participante durante los diez días de prueba.

Posteriormente se registran y analizan los datos que arroja el sistema. Por lo cual, en cada trayectoria se realizan 3 intentos seguidos, obteniendo por día un total de 9 pruebas realizadas. De los cuales, se toma el mejor resultado de los 3 ensayos seguidos. Así como, se muestra en las Tabla 5.1, Tabla 5.2 y Tabla 5.3 Además, la desviación estándar es calculada por medio de la ecuación (5.1), para realizar este cálculo es necesario saber el número de muestras y la media. Estos datos se toman de los resultados de realizar la práctica de cada trayectoria. Para calcular la desviación estándar se debe calcular primero la media teniendo en cuenta la ecuación (5.2). Donde, N es el número de muestras y x_i son los datos que representan a cada participante en la práctica.

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (5.1)$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (5.2)$$

A continuación, se observan las tablas con los datos obtenidos para cada variable en las prácticas realizadas.

Nivel 1											
Día		1	2	3	4	5	6	7	8	9	10
P1	σ	4,6094	7,2148	6,8062	3,0475	7,2031	2,9529	4,0383	5,5321	5,4602	3,2249
	t	11.308	20.693	07.374	05.572	08.572	04.539	06.693	06.473	04.616	07.187
P2	σ	5,7238	18,5087	4,4261	2,8387	3,8723	2,7126	5,3676	5,4338	2,7560	2,2368
	t	17.418	50.066	05.605	05.231	09.000	05.770	08.44	05.82	04.86	05.30
P3	σ	2,4657	4,5651	4,5167	10,6974	6,8231	4,7438	6,7688	2,3420	7,7947	4,9309
	t	06.220	07.715	09.517	12.308	08.649	08.033	48.748	07.110	09.561	03.803
P4	σ	4,1153	3,4808	4,8245	4,5813	1,6820	3,9916	2,7892	1,8077	2,3184	2,7427
	t	02.902	06.968	02.957	02.583	02.407	04.077	02.407	02.044	02.484	02.506
P5	σ	5,0542	5,0995	5,3085	9,2153	5,1691	8,8825	6,1641	2,9913	7,1809	3,2975
	t	01:43.2	04.187	04.176	08.814	06.880	03.418	06.319	04.297	04.803	13.616

Tabla 5.1. Tiempo (mm:ss.ms) y desviación estándar (m.m) nivel 1

Nivel 5											
Día		1	2	3	4	5	6	7	8	9	10
P1	σ	20,0697	6,2576	9,5258	7,8072	5,8623	5,9074	6,4511	4,7716	6,9600	5,3257
	t	30.74	23.077	16.187	14.154	10.946	13.660	07.110	12.616	11.528	10.935
P2	σ	6,3901	8,0224	6,7003	4,6965	5,3852	4,2230	5,5053	4,3470	2,9057	2,6719
	t	30.792	08.858	09.935	12.462	14.869	15.418	09.000	12.869	12.979	18.198
P3	σ	6,4053	7,3226	4,5919	5,2057	7,4415	5,5556	7,3813	5,2094	5,3450	6,7125
	t	21.440	10.044	24.407	11.143	09.935	13.000	24.979	12.121	15.011	11.044
P4	σ	9,0426	8,6585	14,5122	9,3324	8,4471	8,8845	9,1983	5,2914	7,2311	6,1386
	t	21.066	25.572	12.836	09.979	06.198	09.869	07.704	06.407	09.176	15.099
P5	σ	20,6405	9,2646	10,7045	12,7655	11,3108	7,5836	8,5322	9,6692	8,1998	5,7585
	t	03:06.814	08.913	15.319	08.649	14.748	12.429	15.330	16.462	17.913	16.957

Tabla 5.2. Tiempo (mm:ss.ms) y desviación estándar (m.m) nivel 5

Nivel 7											
Día		1	2	3	4	5	6	7	8	9	10
P1	σ	9,6986	8,9721	6,3819	6,4384	6,4319	5,5657	9,8612	9,4737	5,5680	6,1919
	t	18.121	39.044	30.187	25.011	17.836	11.759	11.022	19.253	14.231	15.308
P2	σ	8,3096	19,6191	11,0894	7,9844	5,9984	11,0459	8,6301	6,7545	5,8064	5,1685
	t	31.429	46.671	39.473	11.066	12.935	11.968	21.913	14.605	16.220	14.099
P3	σ	8,7048	9,3966	7,3342	6,9285	5,4686	7,3791	11,2010	6,1834	7,0506	4,5521
	t	18.924	19.440	11.077	11.352	08.066	21.660	22.462	10.011	19.792	16.671
P4	σ	9,6750	7,5716	12,5093	7,2425	8,1659	9,0415	8,1823	9,3499	7,1990	5,6475
	t	18.891	18.055	09.143	09.924	06.616	08.154	13.682	05.671	09.847	16.781
P5	σ	21,9436	15,9771	13,0294	10,9145	14,0509	9,8868	8,8346	10,7769	6,9912	6,8762
	t	40.539	44.176	22.594	17.671	30.143	12.946	11.374	09.253	13.671	08.121

Tabla 5.3. Tiempo (mm:ss.ms) y desviación estándar (m.m) nivel 7

5.4. Análisis de los resultados

Para el análisis de los resultados se muestra gráficamente la evolución del desempeño de los participantes al cabo de los diez días de la realización de la prueba del ambiente virtual. Dicho análisis se realiza teniendo en cuenta las gráficas siguientes, donde se puede observar el comportamiento que tiene la desviación estándar Vs los días de prueba.

La Figura 5.2 muestra el comportamiento de los participantes para los diez días en el nivel 1 siendo este un nivel de menor complejidad. En esta figura se observa que el participante 2 en el día 2 presenta una alta variación en la desviación estándar con un valor por encima de 18 mm comparada con la de

los demás participantes; el comportamiento de los demás participantes en los días restantes no muestra gran variación en la desviación estándar.

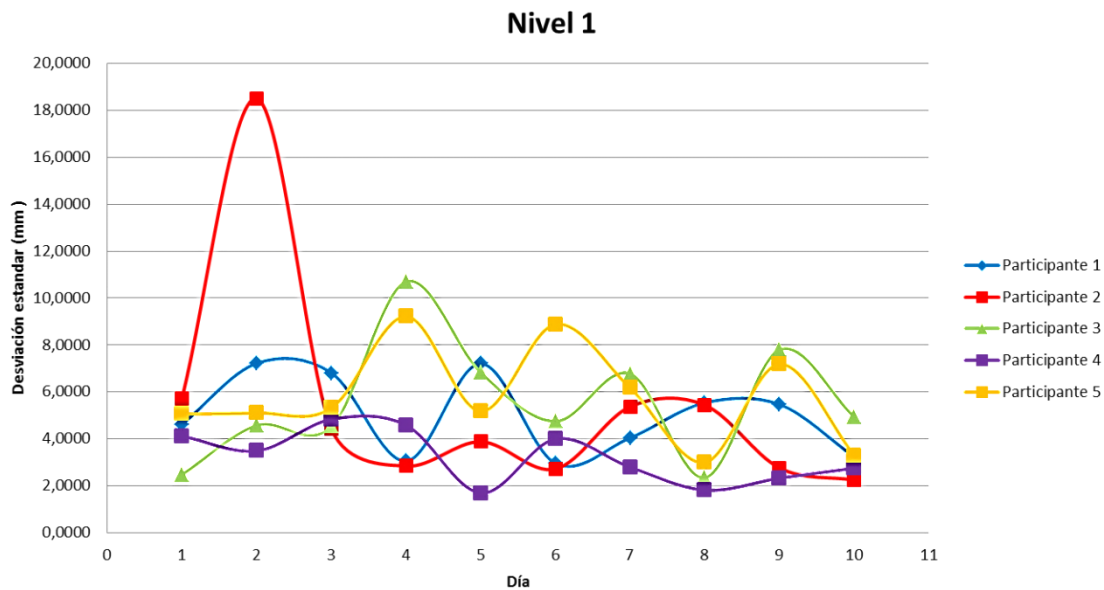


Figura 5.2. Desviación estándar al cabo de diez días para el nivel 1

En la Figura 5.3 se observa que la desviación tiene una tendencia decreciente al pasar de los días; a excepción de los participantes 4 y 5 que presentan una variación contraria para los días 3 y 4 respectivamente.

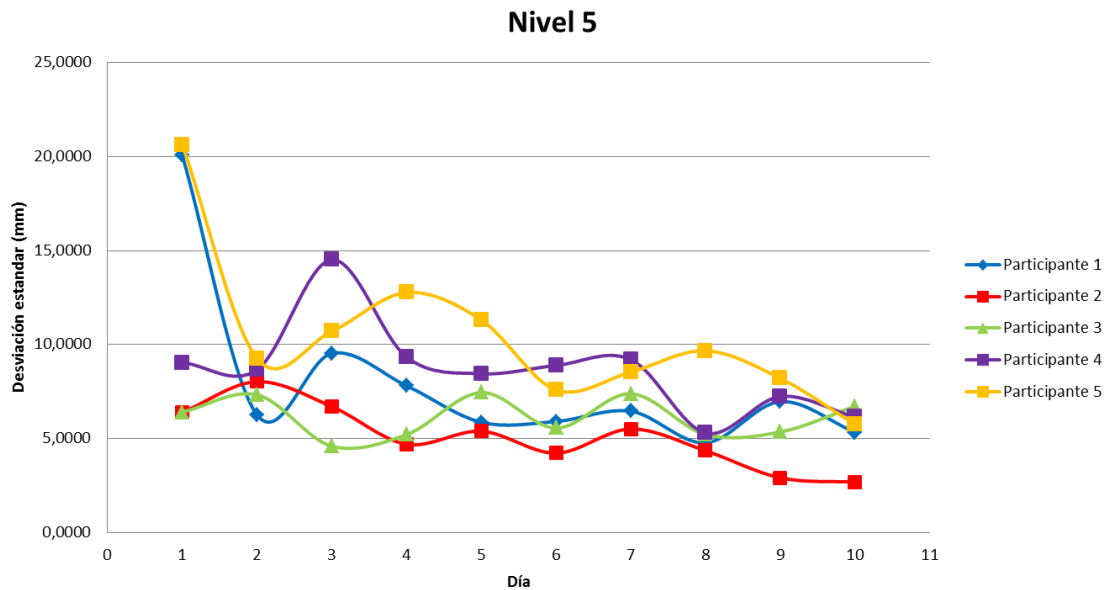


Figura 5.3. Desviación estándar al cabo de diez días para el nivel 5

Finalmente en la Figura 5.4, se observa la evolución del desempeño de los participantes ya que las curvas muestran una tendencia decreciente notoria en la desviación estándar para el nivel 7 (nivel escogido con mayor complejidad). Se puede concluir, que los participantes presentan un buen progreso en cuanto a la adaptabilidad al ambiente y desempeño de la ejecución de los ejercicios.

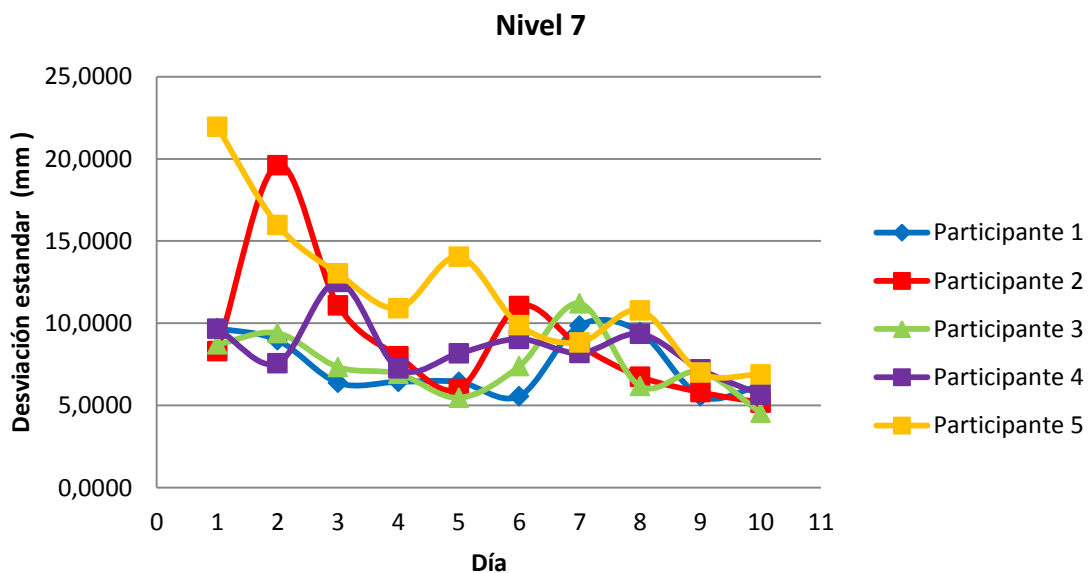


Figura 5.4. Desviación estándar al cabo de diez días para el nivel 7

Para complementar el análisis de los datos obtenidos se calcula la media a los días 1 a 5 y 6 a 10, con el fin de realizar un análisis más completo en cuanto a la adaptabilidad de los participantes al ambiente virtual. Por lo cual, se realizó una inspección comparativa de la media para los 3 niveles escogidos del entorno virtual estos datos fueron consignados en la Tabla 5.4, Tabla 5.5 y Tabla 5.6. En estas, se observa que cada participante disminuye la desviación estándar comparativamente entre los primeros 5 días y los últimos 5 días de prueba. Por tanto, se puede concluir que existe una mejora en cuanto a adaptabilidad de los participantes (disminución del valor de la desviación estándar) a pesar de las fluctuaciones que presenta el participante 3.

Participante	Media días 1 a 5 (mm)	Media días 6 a 10 (mm)
P1	5,7762	4,2417
P2	7,0739	3,7014
P3	5,8136	5,3160
P4	3,7368	2,7299
P5	5,9693	5,7033

Tabla 5.4. Comparación de la Media cinco primeros días y cinco restantes para nivel 1

Participante	Media días 1 a 5 (mm)	Media días 6 a 10 (mm)
P1	9,9045	5,8832
P2	6,2389	3,9306
P3	6,1934	6,0408
P4	9,9985	7,3488
P5	12,9372	7,9487

Tabla 5.5. Comparación de la Media cinco primeros días y cinco restantes para nivel 5

Participante	Media días 1 a 5 (mm)	Media días 6 a 10 (mm)
P1	7,5846	7,3321
P2	10,6002	7,4811
P3	7,5665	7,2733
P4	9,0328	7,8840
P5	15,1831	8,6731

Tabla 5.6. Comparación de la Media cinco primeros días y cinco restantes para nivel 7

CONCLUSIONES

- Se obtuvo la versión del prototipo de un Ambiente Virtual Para Rehabilitación de Motricidad Fina en Miembro Superior Utilizando el Sensor de Movimiento LeapMotion. Este ambiente virtual está compuesto básicamente por una interfaz de usuario que permite el registro y administración de datos basado en lenguaje JAVA; y un ambiente virtual desarrollado en JAVA 3D que está diseñado con el propósito de servir como medio para la realización de las prácticas.
- La estructura del código desarrollado está diseñado en bloques, lo que permite adaptar nuevas características teniendo este código como base de desarrollo de la aplicación esto permite ampliar y mejorar la herramienta para futuros desarrollos.
- Inicialmente se consideró incluir trayectorias 3D prefijadas de distinta complejidad en el ambiente virtual para ser seguidas por el usuario, por lo cual las trayectorias no son precargadas. Ya que se generan aleatoriamente los puntos de control con la elección de nivel de complejidad por parte del usuario. Para lograr el renderizado de la trayectoria de Bézier a través de los puntos de control se utilizan primitivas de las librerías de JAVA 3D.
- En el presente trabajo de grado se puede establecer que la adaptación de los juegos a las interfaces de Rehabilitación de Motricidad Fina en Miembro Superior sirven como medio de ayuda al usuario para realizar actividades que tienen como objetivo trabajar de una manera eficiente la integración viso-motora del humano.
- El entorno virtual se diseñó para facilitar la labor del experto en fisioterapia en cuanto a motricidad fina. Ya que, esta aplicación le permite al profesional llevar un control de los pacientes a tratar y observar el progreso de los mismos por medio de curvas de desempeño.
- Teniendo en cuenta, los resultados obtenidos a lo largo de los 10 diez días considerados como días de prueba para la funcionalidad del ambiente virtual así como se realizó en 5.4 y considerando lo expuesto en la sección 1.3.1 se puede concluir que el entorno virtual diseñado cumple con los aspectos generales de un ambiente virtual como: *participación, interacción e inmersión.*

TRABAJOS FUTUROS Y RECOMENDACIONES

El avance de la tecnología y el desarrollo de entornos virtuales permiten contribuir considerablemente al área de rehabilitación motriz. Se espera, que con el uso del Sensor de Movimiento LeapMotion y partiendo del desarrollo realizado en este trabajo de grado se obtenga una nueva versión que contenga otro tipo de ejercicios o prácticas para trabajar actividades propias de la motricidad fina. Se sugiere que se incluyan otro tipo de ejercicios como: laberintos, juegos relacionados a la rehabilitación, etc., e incluir la incorporación de sonido para mejorar el nivel de inmersión del paciente.

En la versión del prototipo presentado en este trabajo, se considera el ambiente virtual como medio para evaluar el desempeño del paciente en relación a la realización de ejercicios de seguimiento de trayectorias haciendo uso del computador para ejecutar el entorno. Por tanto, se propone que en una nueva versión de la herramienta, se considere adaptar un sistema de seguimiento en línea por medio de una aplicación móvil que permita al fisioterapeuta visualizar el desempeño de los pacientes desde su dispositivo celular.

En el presente trabajo de grado en el estado del arte se mostraron algunos casos de estudio en los que se realizó una investigación científica donde se evidenció la efectividad de los entornos virtuales en rehabilitación motriz. Por tal razón, es viable realizar un estudio clínico con el objetivo de determinar que el aplicativo construido sirve para dicho propósito. Además, este estudio mostraría la eficacia de la herramienta. Por lo tanto es necesario realizar el estudio con el programa de Fisioterapia de la Facultad de Salud con el objetivo de verificar si el Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior es apto para dicho fin y llegar a establecer una calificación de desempeño teniendo en cuenta a la población patológica y no patológica en diferentes rangos de edad.

BIBLIOGRAFÍA

- [1] B. Rogers, H. Bennett, and J. Calabro, *Trastornos de la motricidad*, P. V. H. Services, Ed. New York, Estados Unidos, 200.
- [2] A. Batalla Flores, *Habilidades Motrices*. Barcelona, España: INDE Publicaciones, 2000.
- [3] G. Gualpa Gualán and D. Monge Cruz, "Programa de educación de motricidad fina a través del juego para niños de primero de básica de la escuela fiscal mixta "Isabel Moscoso Dávila"", Universidad del Azuay Tesis, 2010.
- [4] E. M. Cruz Alomoto, "La motricidad fina y su incidencia en el proceso de la pre escritura de los niños del primer año de educación básica del centro educativo "Camilo Gallegos y Soto" de la comunidad chilla chico, cantón saquisilí, provincia de cotopaxi", Universidad Técnica de Ambato Informe Final del Trabajo de Graduación, 2011.
- [5] RAE. (2014, Jun.) Diccionario Real Academia Española. [Online]. <http://lema.rae.es/drae/?val=rehabilitaci%C3%B3n>
- [6] S. Hernández Tápanes, "Conceptos Básicos de Rehabilitación", *infoMED RED DE SALUD DE CUBA*, pp. 1-2, 2014.
- [7] OMS. (2014, Jun.) Organización Mundial de la Salud. [Online]. <http://www.who.int/disabilities/care/es/>
- [8] A. G. Dionísio Corrêa, "Realidade aumentada musical para reabilitação estudo de caso em musicoterapia", Escola Politécnica de Sao Paulo Tesis, 2011.
- [9] M. Greguol Gorgatti and M. T. Silveira Bohme, "Autenticidade científica de um teste de agilidade para indivíduos em cadeira de rodas", *Revista Paulista de Educacao Física*, vol. 17, pp. 41-50, Jun. 2003.
- [10] S. A. D. C. V. REAHBIMEDIC. (2004) REHABIMEDIC. [Online]. http://www.rehabimedic.com/catalogo.html?page=shop.browse&category_id=44

- [11] M. Harms, "Advancing technology in rehabilitation, *Physiotherapy*", *Issue 3*, vol. 98, no. ISSN 00319406, pp. 181-182, Sep. 2012.
- [12] E. Damasceno and J. Junior, "Proposta de um ambiente de reabilitação motora virtual de baixo custo", *6° Workshop de Realidade Virtual e Aumentada*, 2009.
- [13] T. Romero, C. Kirner, and R. Siscoutto, *Fundamentos e Tecnologia de Realidade Virtual e Aumentada*. Belém, PA, 2006.
- [14] M. de Vasconcelos Albeiro, "Um Ambiente Virtual Colaborativo para Treinamento Cirúrgico", Instituto Militar de Engenharia Tesis de Maestria, 2005.
- [15] H. Sveistrup, "Motor rehabilitation using virtual reality", *Journal of Neuro Engineering and Rehabilitation*, pp. 1-8, 2004.
- [16] C. Kirner and T. Kirner, *Virtual Reality and Augmented Reality Applied to Simulation Visualization*. 2008.
- [17] L. Ricci, et al., "A New Calibration Methodology for Thorax and Upper Limbs Motion Capture in Children Using Magneto and Inertial Sensors.", *Sensors*, vol. 14, no. 1, pp. 1057-1072, 2014.
- [18] F. Weichert, et al., "Analysis of the Accuracy and Robustness of the Leap Motion Controller", *Sensors*, vol. 13, pp. 6380-6393, May 2013.
- [19] M. Monroy, et al., "Dispositivos hápticos: una forma de realizar la interacción hombre-máquina", *Robótica y Teleasistencia para Todos*, pp. 39-53, 2007.
- [20] Y. Chang, W. Han, and Y. Tsai, "Kinect-based upper limb rehabilitation system to assist people with cerebral palsy", *Research in Developmental Disabilities*, vol. 34, pp. 3654-3659, 2013.
- [21] J. P. J. León, "Generación de Trayectorias para un Robot PUMA Simulado a partir de Captura de Movimiento de la Mano", Universidad del Cauca Trabajo de Grado, 2013.
- [22] M. H. J. Hijmans, "Bilateral upper-limb rehabilitation after stroke using a movement-based game controller", *Journal of Rehabilitation Research & Development*, vol. 48, no. 8, p. 1005–1014, 2011.

- [23] D. B. e. a. F. Weichert, "Analysis of the Accuracy and Robustness of the Leap Motion Controller", *Sensors*, vol. 13, p. 6380–6393, 2013.
- [24] G. J. e. a. J. Guna, "An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking", *Sensors*, vol. 14, p. 3702–3720, 2014.
- [25] Leap Motion, Inc. (2014, Jun.) Developer Portal Leap Motion . [Online]. <https://developer.leapmotion.com/documentation/>
- [26] C. Sukic and M. Saracevic, "UML and JAVA as Effective Tools for Implementing Algorithms in Computer Graphics", *TEM Journal*, vol. 1, no. 2, pp. 111-116, 2012.
- [27] D. Selman, *Java 3D Programming*, reimpressa, ilustrada ed., Manning, Ed. 2002.
- [28] Inteco, "INGENIERÍA DEL SOFTWARE", INTECO GUÍA DE INGENIERÍA DEL SOFTWARE, 2009.
- [29] J. Joskowicz, "Reglas y Prácticas en eXtremme Programming," University of Manitoba, Manitoba, Canada, Technical report TR, 2008.
- [30] Y. Z. Ordóñez Gómez and C. A. Luna Montilla, "Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior", Universidad del Cauca Trabajo de Grado, 2012.
- [31] J. Lomas Rodríguez, "Estudio e Integración de un Sistema de Control de Entornos Mediante Gestos Usando Kinect", Universidad Autónoma de Madrid, Escuela Politécnica Superior Trabajo de Fin de Grado, 2013.
- [32] I. Hernández Gutiérrez, "Wii-immersion: Aprovechando las Capacidades del Wiimote para la Mejora de la Sensación de Inmersión", UNIVERSITAT POMPEU FABRA, ESCOLA SUPERIOR POLITÈCNICA, ESTUDIS D'INFORMÀTICA Proyecto Fin de Carrera, 2009.
- [33] F. Malouin, et al., "New perspectives of locomotor rehabilitation after stroke", *Medecine Sciences*, pp. 994-998, 2003.
- [34] M. de Vasconcelos Albeiro, "Um Ambiente Virtual Colaborativo para Treinamento Cirúrgico", Instituto Militar de Engenharia Tesis de Maestria, 2005.

- [35] R. Urreola Sanchez, "Integración y/o Confrontación de las Perspectivas Cualitativa y Cuantitativa en Investigación", *Razones Académicas*, no. 2, pp. 11-19, 2011.

ANEXO A

INSTALACIÓN Y CONFIGURACIÓN DEL DISPOSITIVO LEAPMOTION

El dispositivo *LeapMotion* cuenta con un instalador que fue creado con el objetivo de volverla más portable. Además, *LeapMotion* permite ser instalado en 3 sistemas operativos, Mac, Windows y Linux. Se recomienda que para obtener un rendimiento óptimo del dispositivo, se instale en un computador de acuerdo al sistema operativo con las siguientes características:

Mac	Mac® OS X 10.7 Lion Intel® Core™ i3 o superior 2 GB RAM USB 2.0 Puerto Conexión a internet
Windows	Windows® 7 o Windows® 8 AMD Phenom™ II o Intel® Core™ i3 superior 2 GB RAM USB 2.0 Puerto Conexión a internet
Linux	Ubuntu 12 AMD Phenom™ II o Intel® Core™ i3 o superior 2 GB RAM USB 2.0 Puerto Conexión a internet

Así como se muestra en la figura, el kit de LeapMotion contiene:

- Cable USB 2.0
- Manual de instrucción de uso
- Controlador LeapMotion



Figura A 1. Kit de LeapMotion

Para la instalación del dispositivo se deben seguir una serie de pasos de instalación, para que este pueda funcionar correctamente. A continuación los pasos a seguir en la instalación del LeapMotion.

PASO 1: Retirar el adhesivo

Tomando el controlador con las manos se debe despegar la cinta adhesiva que está en la parte superior del LeapMotion. Así como se muestra en la siguiente figura.



Figura A 2. Controlador LeapMotion

De la figura anterior se puede observar que el controlador tiene un adhesivo donde se encuentra impresa la página Web <https://www.leapmotion.com/setup> esta será utilizada para registrar el dispositivo.

PASO 2: Conectar al equipo

Con el cable USB que viene incluido en la caja se debe conectar el dispositivo con el computador en el que se vaya a explorar el controlador. Además, se debe tener en cuenta al momento de conectarlo que el lado brillante del dispositivo debe quedar hacia arriba y la luz verde que se enciende al conectarlo al computador debe mirar hacia el usuario. Tal como se muestra en la siguiente Figura



Figura A 3. Conectar el controlador al equipo. Imagen tomada de: <https://www.leapmotion.com/setup>

PASO 3: Ubicar el dispositivo

Se debe ubicar el controlador frente al usuario, para captar de manera efectiva los movimientos se sugiere que el usuario debe apoyar los codos en la mesa. La ubicación del dispositivo respecto al usuario se muestra en la siguiente figura.



Figura A 4. Ubicar el controlador. Imagen tomada de: <https://www.leapmotion.com/setup>

ANEXO B

PROCEDIMIENTO DE INSTALACIÓN DE LA APLICACIÓN CON TODOS SUS REQUERIMIENTOS

La herramienta desarrollada cuenta con un archivo ejecutable que fue creado con el objetivo de ser instalado en otros equipos de una manera fácil y sencilla. Se recomienda que para obtener un rendimiento óptimo de la herramienta, se instale en un computador con las siguientes características:

Sistema Operativo: Windows 7.

Memoria RAM: 2 GB o superior.

Preferiblemente con tarjeta aceleradora de video.

Procesador: Intel Core i3 o superior.

Puerto USB

Antes de proceder con la instalación de la herramienta, se debe instalar los drivers del dispositivo LeapMotion, para que este funcione correctamente. Tal como se describe en el **ANEXO A**. A continuación, se muestran los pasos que se deben seguir para la instalación del Ambiente virtual para Rehabilitación de Motricidad Fina en Miembro Superior.

NOTA: Para proceder con la instalación de la herramienta, el equipo debe tener instalado la máquina virtual de java que por defecto viene instalado, de lo contrario debe ir a la página web: www.java.com/es/download descargar e instalar los componentes, además, se necesita el gestor de base de datos de MySQL que se puede encontrar en la página web: www.mysql.com/downloads.

Paso 1: Ejecutar el instalador ReMoFi-Setup.exe

Iniciando, en la carpeta de instalación de la aplicación se encuentra el archivo **ReMoFi-Setup.exe**. Al ejecutar el instalador, se abre una ventana (Figura B. 1)

de instalación donde se debe elegir el idioma, debe oprimirse aceptar y continuar en la bienvenida a la instalación.

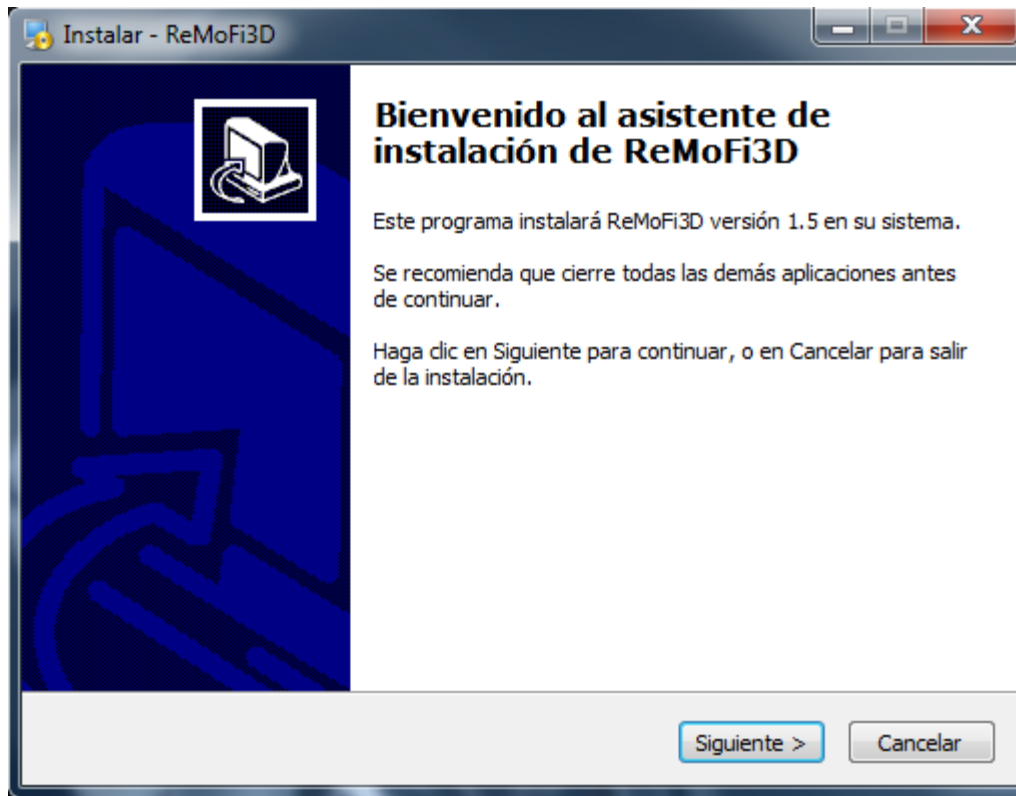


Figura B. 1 Ventana de instalación. Fuente propia

En la siguiente ventana (Figura B. 2), se proporciona la ruta (C:\Program Files (x86)\ReMoFi3D) por defecto donde se copiarán los componentes necesarios para la ejecución, también se permite anclar un acceso rápido en Menú inicio de Windows y en el escritorio (Figura B. 3). Después de copiar y realizar los procesos necesarios se de finalizar la instalación.

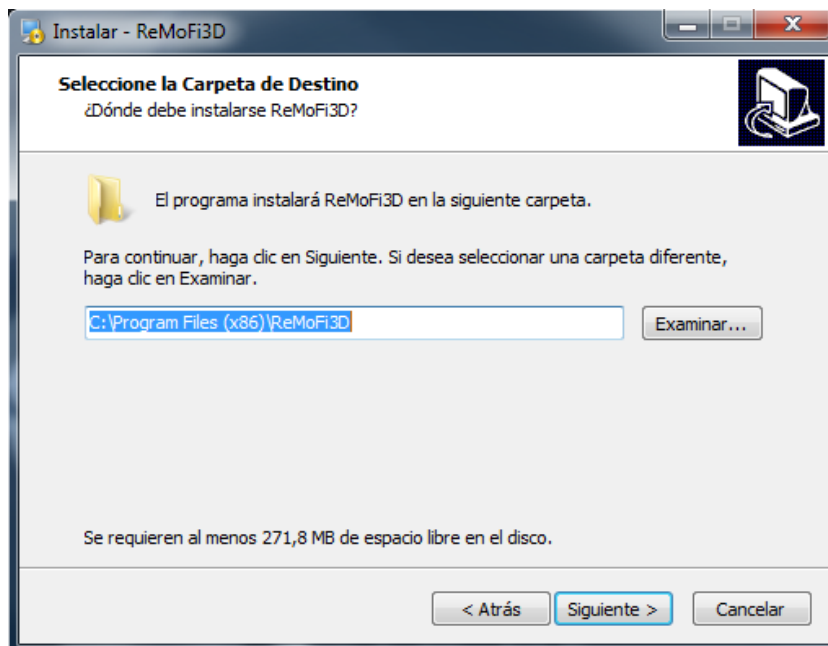


Figura B. 2. Elección de carpeta de instalación. Fuente propia

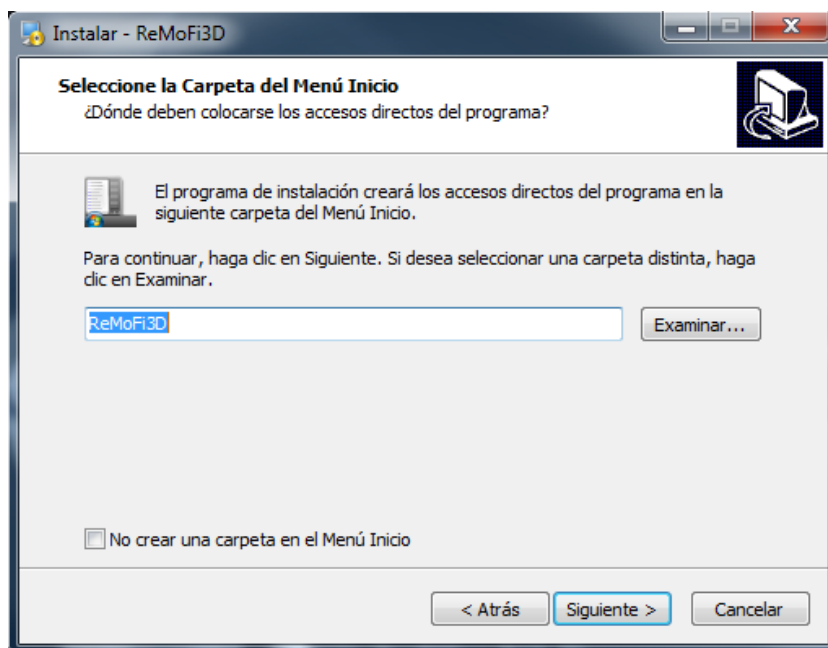


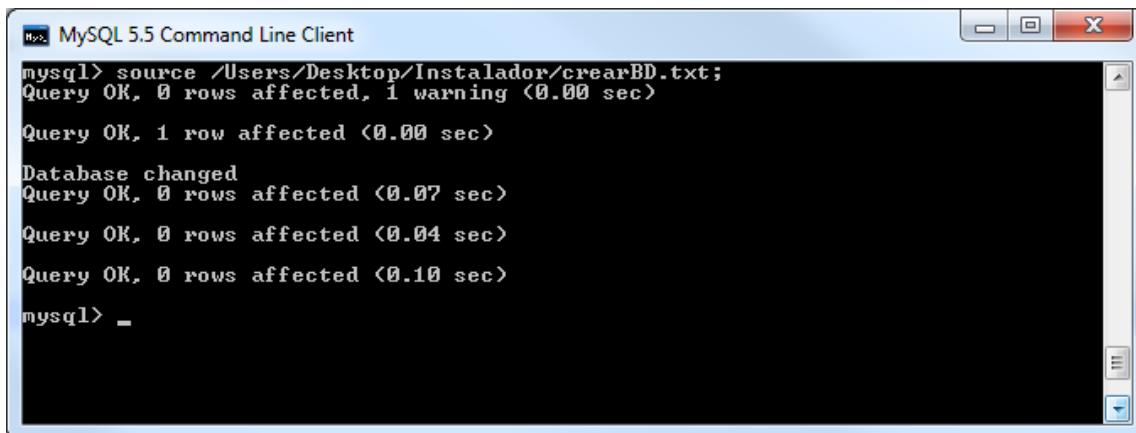
Figura B. 3. Accesos rápidos para la Aplicación. Fuente propia

Una vez concluido este paso ya se tendrá instalada la aplicación, pero antes de ejecutarla debe configurarse la base de datos (Paso 2), de lo contrario la aplicación no tendrá un funcionamiento adecuado.

Paso 2: Conexión a la base de datos

Antes de iniciarse por primera vez la herramienta se debe crear la base de datos o importar una base de datos ya establecida. Esto con el fin de guardar los registros de pacientes o inspeccionar datos de pacientes ya registrados en una base de datos.

Buscar MySQL en el menú inicio y abrir MySQL Command Line Cliente, una vez que cargue la consola, debe ejecutar el comando *source*, seguido de la ruta de ubicación del archivo script llamado crearBD.txt que se encuentra dentro de la carpeta de instalación. Como se observa en la Figura B. 4, al ejecutar el comando “mysql> SOURCE /Users/Desktop/Instalador/crearBD.txt;” debe crear correctamente todos los campos necesarios y a continuación se puede continuar con la ejecución de la herramienta.



```
mysql> source /Users/Desktop/Instalador/crearBD.txt;
Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.07 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.10 sec)

mysql> _
```

Figura B. 4. Crear Base de Datos por primera vez. Fuente propia

NOTA: La versión para desarrolladores contiene una carpeta ReMoFi-install-Develop donde se encuentran diez archivos numerados por pasos para instalar.

