

Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior



Tesis presentada a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del
Título de:

Ingeniero en Automática Industrial

Yamile Zuleima Ordóñez Gómez
Carlos Andrés Luna Montilla

Director:
PhD. Carlos Felipe Rengifo Rodas

Popayán, Diciembre de 2012

Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior



Yamile Zuleima Ordóñez Gómez
Carlos Andrés Luna Montilla

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Línea de I+D en Robótica y Control
Ingeniería en Automática Industrial

Popayán, Diciembre de 2012

Tabla de contenido

| | |
|--|----|
| ÍNDICE DE FIGURAS | 3 |
| ÍNDICE DE TABLAS | 6 |
| INTRODUCCIÓN | 7 |
| DESCRIPCIÓN DEL TRABAJO REALIZADO | 8 |
| ORGANIZACIÓN DE LA MONOGRAFÍA | 11 |
| CAPÍTULO 1 | 12 |
| CONCEPTOS PRELIMINARES..... | 12 |
| 1.1 Conceptos sobre habilidades motrices y motricidad fina | 12 |
| 1.2 Conceptos de Rehabilitación | 13 |
| 1.3 Realidad virtual. | 16 |
| 1.4 Sistemas Hápticos | 20 |
| 1.5 Aplicación de realidad virtual en terapias de rehabilitación de motricidad fina. | 30 |
| CAPÍTULO 2 | 34 |
| HERRAMIENTAS DE SOFTWARE Y DISPOSITIVO HÁPTICO EMPLEADO | 34 |
| 2.1 Selección de herramientas Software | 34 |
| 2.2 Dispositivo Omni Phantom..... | 35 |
| 2.3 OpenHaptics Toolkit..... | 37 |
| 2.4 Desarrollo de una aplicación háptica usando el dispositivo Omni Phantom. | 42 |
| CAPÍTULO 3 | 48 |
| TRAYECTORIAS Y CAMPO DE FUERZA..... | 48 |
| 3.1 Elección del método de trazado de trayectorias | 49 |
| 3.2 Curvas de Bézier para las trayectorias de la aplicación..... | 54 |
| 3.3 Apariencia 3D de las trayectorias | 57 |
| 3.4 Implementación del modelado matemático de las trayectorias en el ambiente virtual | 58 |
| 3.5 Campo de fuerza de asistencia al seguimiento de las trayectorias | 60 |

| | |
|--|----|
| CAPÍTULO 4 | 68 |
| DESCRIPCIÓN DEL PROTOTIPO DESARROLLADO | 68 |
| 4.1 Descripción de la Interfaz de Usuario | 68 |
| 4.2 Ambientes Virtuales de Entrenamiento | 76 |
| CONCLUSIONES..... | 83 |
| TRABAJOS FUTUROS Y RECOMENDACIONES..... | 85 |
| BIBLIOGRAFÍA | 86 |
| ANEXO A | 89 |
| PRESENTACIÓN DE HERRAMIENTAS COMPLEMENTARIAS | 89 |
| ANEXO B | 94 |
| MODELADO DE LAS TRAYECTORIAS EN UN PROGRAMA CAD Y SU ADECUACIÓN PARA SER IMPORTADAS MEDIANTE OPENGL..... | 94 |
| ANEXO C | 97 |
| INSTALACIÓN DEL PROTOTIPO AMBIENTE VIRTUAL PARA REHABILITACIÓN DE MOTRICIDAD FINA EN MIEMBRO SUPERIOR. | 97 |

Índice de figuras

| | |
|---|----|
| Figura 1.1. Imagen de visita virtual, Acrópolis, 400 A.C Enciclopedia Microsoft Encarta 2009..... | 18 |
| Figura 1.2. Caverna virtual | 19 |
| Figura 1.3. Comparación del flujo de información en una aplicación de RV empleando un dispositivo háptico y un dispositivo convencional (Elaboración Propia)..... | 21 |
| Figura 1.4. Fuerza de restauración, representada gráficamente..... | 24 |
| Figura 1.5. Tipos de fricción. | 25 |
| Figura 1.6. Guantes fabricados por Immersion Cooperation..... | 27 |
| Figura 1.7. Principales dispositivos hápticos disponibles en el mercado. | 29 |
| Figura 1.8. Prototipo del sistema GENTLE/S. | 30 |
| Figura 1.9. Entornos de entrenamiento en [6]. | 31 |
| Figura 1.10. Trayectorias del ambiente bidimensional descrito en [7]..... | 31 |
| Figura 1.11. Laberinto utilizado para medición de habilidades motrices en [8]. | 32 |
| Figura 1.12. Esquema del sistema detallado en [9]..... | 33 |
| Figura 2.1. Dispositivo háptico Omni Phantom..... | 35 |
| Figura 2.2. Indicación de articulaciones del dispositivo háptico Omni Phantom | 36 |
| Figura 2.3. Arquitectura de biblioteca OpenHaptics Toolkit | 37 |
| Figura 2.4. Funcionalidades de la librería HDAPI..... | 41 |
| Figura 2.6. Estructura de una aplicación háptica..... | 43 |
| Figura 3.1. Recorrido que realiza $B(s)$ cuando la curva de Bézier es de primer orden. | 53 |
| Figura 3.2. Recorrido que realiza $B(s)$ cuando la curva de Bézier es de segundo orden. | 53 |
| Figura 3.3. Puntos de control e intermedios para la construcción de una curva de Bézier de cuarto orden. | 54 |
| Figura 3.4. Distribución de los vértices de control en el plano xy. | 55 |
| Figura 3.5. Polígono de control de la curva de Bézier del Nivel 1 de entrenamiento. | 56 |

| | |
|--|----|
| Figura 3.6. Curva de Bézier con su respectivo polígono de control. | 57 |
| Figura 3.7. Ambiente Virtual de entrenamiento. | 58 |
| Figura 3.8. Representación del sistema de resorte encargado de aplicar la fuerza. | 64 |
| Figura 3.9. Posibles direcciones de la fuerza ejercida por el dispositivo háptico. | 65 |
| Figura 3.10. Vector 1: punto sobre la trayectoria. Vector 2: posición del cursor háptico. Vector 3: diferencia. | 65 |
| Figura 4.1. Maneras de acceder a formularios de la aplicación | 69 |
| Figura 4.2. Mensaje de alerta generada por campos vacíos. | 69 |
| Figura 4.3. Interfaz de Usuario, Formulario uno, 'Registro Pacientes' | 70 |
| Tabla 4.1. Dimensiones de la Motricidad Fina [10]. | 71 |
| Figura 4.4. Selección de ítem y respuesta a actividades | 72 |
| Figura 4.6. Mensaje de alerta generado por número de documento repetido .. | 73 |
| Figura 4.5. Interfaz de Usuario, Formulario dos, 'Consultar Registro' | 73 |
| Figura 4.7. Mensaje generado por introducción de documento no registrado.. | 74 |
| Figura 4.8. Botón y opción empleada para la consultar una evaluación | 74 |
| Figura 4.10. Acceso a los entornos virtuales. | 76 |
| Figura 4.11. Ventana de ejercicios de familiarización. | 77 |
| Figura 4.12. Trayectorias de ejercicios de familiarización. | 78 |
| Figura 4.13. Ventana del ambiente virtual correspondiente al Nivel 1 de entrenamiento. | 79 |
| Figura 4.14. Mensajes y colores del toroide dependientes de la desviación del mismo respecto a la trayectoria. | 80 |
| Figura 4.15. Sub-ventana con los datos diligenciados. | 81 |
| Figura 4.16. Trayectorias para los cinco niveles de entrenamiento. | 82 |
| Figura A.1. Estructura librería Qt. | 91 |
| Figura B.1. Selección del entorno "Conjunto", es Solid Edge. | 94 |
| Figura B.2. Selección de la aplicación para crear tuberías. | 95 |
| Figura B.3. Gráfica de la curva de Bézier para el nivel 1 en Matlab. | 95 |
| Figura B.4. Izquierda, boceto perfil de dibujado de la herramienta XpresRoute. Derecha, trayectoria en 3D implementada en Solid Edge. | 96 |
| Figura B.5. Ambiente de trabajo del programa Blender. | 96 |
| Figura C.1. Ubicación del instalador. | 98 |

| | |
|--|-----|
| Figura C.2. Cuadro de diálogo inicial de instalación..... | 98 |
| Figura C.3. Selección de ubicación de archivos..... | 99 |
| Figura C.4. Edición de los permisos de la carpeta que contiene el ejecutable de la aplicación..... | 100 |

Índice de tablas

| | |
|---|----|
| Tabla 1.1. Características principales de los ambientes virtuales. | 16 |
| Tabla 1.2. Aspectos considerados en Realidad Virtual | 19 |
| Tabla 1.3. Funciones que llaman a propiedades de materiales usando dispositivo háptico Omni Phantom | 26 |
| Tabla 1.4. Principales Fabricantes y característica mas importantes de dispositivos hapticos disponibles en el mercado | 28 |
| Tabla 2.1. Especificaciones técnicas del dispositivo Omni Phamton. | 36 |
| Tabla 2.2. Funciones principales de la API HDAPI | 39 |
| Tabla 2.3. Funciones principales de la API HLAPI | 40 |

Introducción

La rehabilitación motriz es un proceso que busca la mejora de funciones que se han perdido o se han visto afectadas por alguna patología o lesión. El objetivo general de un proceso de rehabilitación es mejorar el desempeño de las actividades diarias, lo que permite a los pacientes adquirir un alto grado de independencia y mejorar la calidad de sus vidas. Tres factores pueden incidir directamente en la recuperación motriz; la intervención temprana, la realización de ejercicios orientados a la ejecución de tareas y la repetición e intensidad con que se realicen [1].

Una de las primeras labores en el proceso de rehabilitación, es identificar los medios que podrían admitir la realización de tareas repetitivas que incluyan diferentes modalidades sensoriales, como visión, propiocepción y audición, entre otras; que permitirán el mejoramiento de determinadas funciones. Los ambientes virtuales (AV) son una buena opción para potenciar y agilizar dichas actividades. Los AVs son escenarios gráficos, en los que el usuario interactúa con entornos tridimensionales, en tiempo real, haciendo uso de dispositivos multi sensoriales [2]. El objetivo de estos ambientes es sumergir al usuario en un mundo virtual, donde se sustituyen los estímulos provenientes del entorno real por estímulos sintéticos. De la combinación de rehabilitación y realidad virtual, resulta el concepto de rehabilitación virtual, que es el entrenamiento basado en ejercicios de simulación mediante tecnología de realidad virtual.

La rehabilitación virtual permite a profesionales en la salud estimular el sistema sensorial de manera precisa y adecuada a través de la realización de ejercicios controlables, repetibles y seguros; dichos ejercicios son creados con un trasfondo lúdico enfocado al propósito de mejorar la motivación de los pacientes. La rehabilitación virtual también ofrece la posibilidad de individualizar las necesidades de tratamiento al tiempo que proporciona una mayor estandarización de los protocolos de evaluación y entrenamiento.

En [3], se muestra un estudio experimental en el cual se comparan ejercicios de rehabilitación convencionales y ejercicios basados en realidad virtual; se

evidencia que la realidad virtual ofrece la posibilidad de que las terapias puedan ser proporcionadas dentro de un contexto funcional, útil y motivador; y que además puedan ser fácilmente clasificadas y documentadas.

El éxito de la rehabilitación virtual, depende en gran medida del nivel de inmersión que el paciente alcance sobre el ambiente virtual; las interfaces hápticas o kinestésicas podrían mejorar significativamente dicha inmersión. Este tipo de interfaces son dispositivos bidireccionales que realimentan señales táctiles y propioceptivas a un usuario, sobre información de objetos que se encuentren dentro del ambiente virtual; de este modo el usuario podría percibir las características propias de los objetos, como textura, forma y rigidez. Las interfaces hápticas comprenden dispositivos como guantes, joysticks y exoesqueletos, los cuales cuentan con retroalimentación de fuerza para la simulación de contacto con objetos sólidos.

La efectividad de la incorporación de dispositivos hápticos en sistemas de rehabilitación virtual, se han evidenciado en distintos estudios, por ejemplo en [4], se concluye que los dispositivos hápticos mejoran el rendimiento de los pacientes, cuando estos son sometidos a realizar tareas de alta complejidad; y en [5], se muestran mediciones objetivas y subjetivas que indican que la información háptica contribuye a la mejora de funciones en los miembros superiores, en pacientes que sufrieron un accidente cerebro-vascular.

Algunos de los trabajos relacionados con rehabilitación virtual enfocados hacia rehabilitación motriz en miembro superior y que tratan tareas propias de la motricidad fina, son básicamente ambientes virtuales, en los que los pacientes interactúan haciendo uso de dispositivos hápticos; generalmente utilizan alguno de los dispositivos de Sensable Technologies como la interfaz Omni Phantom y la Phantom Premium; aunque existe un caso en el que se utiliza la interfaz Haptic Master que tiene la capacidad de generar una fuerza mucho mayor y se utiliza para sistemas de rehabilitación en los que se realiza un entrenamiento que incluye la totalidad del miembro superior [6].

Descripción del trabajo realizado

En el presente documento se describe el desarrollo de una aplicación para rehabilitación virtual basada en el dispositivo háptico Omni Phantom. Esta aplicación permite la realización de ejercicios de motricidad fina con grado de dificultad variable. Además de la implementación del ambiente virtual donde se realizan los ejercicios, el aplicativo desarrollado permite administrar la información de los pacientes que los utilizan así como valores cuantitativos de los resultados obtenidos en cada sesión de entrenamiento. Es importante recalcar que este primer trabajo no comprende el estudio clínico.

En el ambiente virtual descrito en este trabajo, el órgano terminal de la interfaz háptica es representado como un anillo. El objetivo del ejercicio consiste en centrar dicho anillo alrededor de una trayectoria de curvatura variable. Un elemento novedoso de la aplicación propuesta es que la trayectoria a seguir puede ejercer una fuerza de atracción sobre el centro del anillo. La intensidad de la fuerza de atracción puede ser configurada al inicio del ejercicio, esto incide en el grado de dificultad del seguimiento de la trayectoria.

A diferencia de las aplicaciones presentadas en [7], [8] y [9]; en esta, la fuerza de atracción es un vector en el espacio y no en el plano. Otro elemento diferenciador es que las trayectorias a seguir no presentan cambios bruscos de dirección. Por el contrario, son curvas suaves lo que se traduce en una mayor comodidad al momento de seguirlas. En [6], [7] y [8], el cursor háptico es representado por una esfera, la cual no permite aprovechar los seis grados de libertad de la Phantom (3 relacionados con posición y 3 relacionados con orientación). En [6], se utiliza un dispositivo de tres grados de libertad y no de seis, para la cual resulta perfecta la utilización de una esfera como cursor háptico. En el presente trabajo se usan los seis grados de libertad de la interfaz háptica, debido a la forma de las trayectorias y a la del objeto 3D que se carga como cursor háptico; esto permite entrenar en el miembro superior no sólo la muñeca sino también los dedos, lo cual contribuye a mejores prácticas para rehabilitación de la motricidad fina y coordinación mano-ojo.

En los trabajos descritos en [7], [8] y [9], se obtienen mediciones que se asocian a un paciente en tratamiento; sin embargo, en ninguna de ellas se menciona la forma en la que se administran los datos del paciente o la manera en la que se asignan estos datos una vez se haya terminado el tratamiento. Al parecer en todas estas aplicaciones sólo se tiene la ventana correspondiente al ambiente virtual en el que se interactúa con el dispositivo háptico y en algunos casos se tienen subventanas para monitorear las variables de este ambiente.

Cuando se realizan varias prácticas en un mismo ambiente virtual y se hacen mediciones que se asocian a un paciente, resulta oportuno tener una base de datos que administre esta información y así poder consultarla en el momento que sea necesario; es por eso que en la aplicación desarrollada y descrita en este trabajo Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior, antes de entrar en la ventana del ambiente virtual de entrenamiento, se tiene una interfaz de usuario que le permite al encargado de supervisar y guiar las prácticas, hacer un registro del paciente que se va a entrenar; de esta manera, al momento de realizar la práctica, se puede cargar el registro del paciente en cuestión y a este se le asocia el puntaje obtenido cuando haya finalizado la práctica. De esta manera se pueden manejar muchos

registros de pacientes sin temor a que la información entre estos se asigne erróneamente.

Uno de los retos que representó la realización de este trabajo fue el de la implementación de las trayectorias curvas en el ambiente virtual, pues en los trabajos descritos en [6], [7] y [8], las trayectorias son del mismo tipo, porque su modelado es relativamente fácil y consiste en hacerlo tramo a tramo mediante líneas rectas, una seguida de otra, que pueden estar en sentido horizontal o vertical. Después de analizar varios métodos se eligió el de curvas de Bézier, pues permite la creación trayectorias curvas, y no exige demasiados recursos computacionales para recalcularlas durante la ejecución de la aplicación. Estas curvas son construidas a partir de puntos distribuidos en el plano que gobiernan su dirección, ya que ésta no toca dichos puntos a excepción del punto inicial y el final.

La implementación del campo de fuerza también acarrió con sus propias demandas; en primer lugar, fue necesario realizar la programación del dispositivo háptico a bajo nivel para que fuera posible personalizar las fuerzas que este genera y dejar a un lado la manera predeterminada de funcionamiento; en segundo lugar se establece el método para calcular la magnitud y orientación de la fuerza en el espacio, que se aplica en un determinado instante de tiempo. Para ello, un algoritmo relaciona el modelado de la trayectoria curva con la posición que se lee del dispositivo háptico. Esta posición entra en un ciclo que calcula las distancias hacia cada uno de los puntos que conforman la trayectoria; se elige el punto que se encuentra más cercano a la posición del cursor háptico y se realiza una diferencia vectorial entre los dos (posición del cursor háptico y punto más cercano sobre la trayectoria); al normalizar el resultado se obtiene la dirección en la que se debe aplicar la fuerza. El valor del que depende la magnitud de esta fuerza es la distancia más corta calculada previamente (la magnitud de la fuerza de atracción se define como proporcional a la distancia entre el centro del anillo y la trayectoria a seguir). Otros factores que se incluyen en el cálculo de la fuerza, son el grado de asistencia ingresado por el fisioterapeuta a través de la interfaz de usuario del ambiente virtual, cuyo valor puede establecerse entre cero y uno; y un factor delimitador de la fuerza generada por el dispositivo para garantizar la seguridad del mismo. Este último factor delimitador es necesario porque el dispositivo no está programado con las librerías de alto nivel que tienen establecidos los límites admisibles de las fuerzas generadas.

La distancia mínima entre la posición del cursor háptico y los puntos de la trayectoria es utilizada también en el método que proporciona una forma de medir el desempeño del paciente en las prácticas. Este método consiste en promediar la desviación (distancia mínima); medida desde el cursor háptico hasta el cuerpo de la trayectoria que se desea seguir. El algoritmo que calcula

esta desviación promedio, cuida que en el cálculo realizado no se ingresen datos indeseados como la repetición de una medida de desviación cuando el paciente mantiene el cursor en un solo punto, ya que si se ingresaran todos estos datos repetidos, el promedio que se calcula perdería su objetividad y no representaría una forma confiable de la medición del desempeño. Cuando el paciente termina el recorrido de la trayectoria con el cursor háptico, el campo de fuerza se reduce al vector de fuerza antigraavedad; en este punto el cálculo de la desviación promedio también se ha detenido y el valor de éste se encuentra listo para ser almacenado en la base de datos y asignarse al registro del paciente que acaba de realizar el entrenamiento.

Organización de la monografía

El presente documento está compuesto por cuatro capítulos. En el Capítulo 1, se detallan los conceptos relativos a habilidades motoras, rehabilitación motriz, realidad virtual y sistemas hápticos. El objetivo de este capítulo es proveer un acercamiento hacia los conceptos que abarca el prototipo Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior.

En el Capítulo 2, se exponen las herramientas de software y hardware que permitieron la consecución de los resultados expuestos al final de este trabajo. El dispositivo háptico utilizado para interactuar en el ambiente virtual fue el OmniPhantom de Sensable Technologies; por lo tanto en el capítulo II se hace énfasis en la descripción de este dispositivo y en la creación de aplicaciones hápticas usando las librerías para la programación del mismo, suministradas por su distribuidor.

El método de curvas de Bézier mediante el cual fueron generadas las trayectorias del ambiente virtual, así como las operaciones que llevan a la generación del campo de fuerza de asistencia en el entrenamiento, el algoritmo que permite medir el desempeño de los pacientes y la adecuación del ambiente virtual de entrenamiento, se detallan en el Capítulo 3.

En el Capítulo 4, se describe el funcionamiento de la interfaz de usuario y del ambiente virtual de entrenamiento. Esta información permitirá al encargado de supervisar las prácticas y realizar las evaluaciones de los pacientes, tener conocimiento del procedimiento que se sugiere seguir para hacer un buen uso de la aplicación en su totalidad.

Finalmente se presentan las conclusiones, recomendaciones y los trabajos que podrían desarrollarse en el futuro, basándose en los conocimientos adquiridos mediante la realización de este trabajo de grado, sobre ambientes virtuales con realimentación visual y de fuerza.

Capítulo 1

Conceptos preliminares

El control de la motricidad fina puede ser definido como la capacidad de coordinación de músculos, huesos y nervios para producir movimientos pequeños y precisos. Son múltiples las causas que originan problemas de motricidad. Pueden ir desde degeneraciones neurológicas, accidentes vasculares cerebrales hasta lesiones por esfuerzos repetitivos y cotidianos, o simplemente podrían deberse a un desarrollo no adecuado. Todos estos problemas requieren intervención médica con el fin de rehabilitar al paciente.

En los avances de la computación aplicada a la salud, es cada vez más común el uso de la realidad virtual asociada al tratamiento de rehabilitación motriz, sustituyendo poco a poco los procedimientos convencionales de tratamiento. La realidad virtual hace posible que las actividades propias del proceso de rehabilitación estén enmarcadas dentro de un entorno cómodo, motivador e intuitivo, además de que permite que estas actividades sean adaptables y personalizadas para cada paciente.

En este capítulo se presentan algunos conceptos que involucra la rehabilitación virtual enfocada a motricidad fina. Se describen los aspectos generales referidos a rehabilitación motriz y realidad virtual. Debido a que en el desarrollo de la aplicación descrita en este trabajo se emplea el dispositivo háptico Omni Phantom para la interacción del paciente con el ambiente virtual, se describen en detalle los conceptos propios de los sistemas hápticos. Finalmente, se muestran algunas aplicaciones relacionadas con los propósitos de este trabajo.

1.1 Conceptos sobre habilidades motrices y motricidad fina

Las habilidades de motricidad son las que permiten a las personas realizar funciones de movilidad y locomoción, así como las de mantener la postura del

cuerpo. También aquellas habilidades que posibilitan a los individuos relacionarse con el mundo y los objetos, reconocerlos, establecer contacto con ellos y llegar a dominarlos dotándolos de funcionalidad [10].

La adquisición de habilidades motrices resulta de una combinación de distintos sistemas de regulación: el de las estructuras perceptuales, que permiten el reconocimiento de características del ambiente y del cuerpo; las habilidades cognitivas, que permiten la formulación de planos simbólicos u objetivos en relación al ambiente actual y a la experiencia pasada; y el sistema que efectúa el movimiento, compuesto por músculos y articulaciones que deben garantizar la elasticidad y/o compresión y estabilidad para realizar movimientos coordinados [11].

Las habilidades de motricidad se dividen en dos grandes grupos: habilidades relativas a la motricidad gruesa y habilidades referidas a la motricidad fina.

Las habilidades de motricidad gruesa son aquellas en las que se incluyen movimientos generales del cuerpo. Estos movimientos son contracciones coordinadas de los grandes músculos del cuerpo que acontecen al sentarse, al andar, al lanzar objetos como una pelota, al subir y bajar escaleras.

Entre tanto las habilidades de motricidad fina son una destreza o sistema digito-manual que tiene por finalidad coordinar los músculos finos de la mano, en la manipulación de instrumentos y materiales que requieren de mucha precisión al momento de realizar algunas actividades como prensar, coger, ensartar, atar y manipular utensilios tales como el lápiz. Es gracias a esta habilidad que se puede lograr la ejecución de una escritura clara y precisa, al momento de hacer una representación gráfica de símbolos y signos a través de trazos [12].

Un desarrollo no adecuado o alteración de motricidad fina, entre otras cosas conllevan a problemas con la escritura o trazos previos a ésta, más conocidos como grafo motricidad, y son más frecuentes en los primeros años de vida. Hay cinco componentes identificados que contribuyen al buen desempeño de la motricidad fina y por ende a la calidad de la escritura: cinestesia, la planificación motora, coordinación ojo-mano, la integración viso-motora, y la manipulación de la mano [7]. 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

La palabra rehabilitación se deriva del término habilitar, del latín *habilitare*, verbo transitivo que significa volver hábil, capaz, o apto. Con la inclusión del

prefijo “re”, re-habilita toma el sentido de adquirir nuevamente una habilidad perdida o disminuida [13].

De acuerdo a la OMS (Organización Mundial de la Salud) se define rehabilitación como “parte de la asistencia médica encargada de desarrollar las capacidades funcionales y psicológicas del individuo y activar sus mecanismos de compensación, a fin de permitirle llevar una existencia autónoma y dinámica”.

En el área médica, el concepto de rehabilitación siempre asume una connotación terapéutica, puesto que busca devolver al individuo, con algún tipo de incapacidad, el máximo de funcionalidad posible [14].

Las actividades de rehabilitación motora buscan mejorar, mantener o frenar las deficiencias motrices que normalmente son caracterizadas por disturbios en la estructura o funciones del cuerpo que interfieren en el movimiento y/o locomoción del individuo [15]. Dichas funciones podrían limitar o afectar el desempeño de las personas en la ejecución de labores en su cotidianidad, así pues la rehabilitación se considera un proceso que resulta de la aplicación integrada de muchos procedimientos para lograr que los individuos recuperen su estado funcional óptimo. Existen varios tipos de rehabilitación, entre los cuales están: rehabilitación física, rehabilitación cognitiva, musicoterapia, práctica mental y rehabilitación virtual. Aunque estos tipos terapias emplean técnicas y elementos distintos para la consecución de sus objetivos, todas coinciden en el propósito de estimular las funciones mentales que incluyen la atención, percepción, comprensión, aprendizaje, memoria y raciocinio entre otros, estas funciones propias de la cognición son requeridas para la realización de secuencias motoras complejas necesarias en la ejecución de actividades como tocar instrumentos musicales, dibujar y escribir. Debido a los propósitos generales del desarrollo del presente trabajo, en este documento se aborda rehabilitación virtual y todos los aspectos relacionados con la misma.

En [16] se sugiere que en los procesos de rehabilitación, la realización constante de ejercicios que involucran habilidades motrices específicas conlleva a la mejora en el desempeño de algunas tareas. Los ejercicios deben ser realizados en ambientes adecuados que incluyan diferentes modalidades sensoriales.

Las herramientas computacionales pueden contribuir significativamente en al menos dos aspectos de los procesos de rehabilitación motriz, el primero referente al almacenamiento y administración de información (sobre el paciente, enfermedad, proceso de rehabilitación y del terapeuta), el segundo en servir como medio que permite llevar a cabo ejercicios repetitivos que incluyan distintos sentidos como visión, audición, propiocepción.

1.2.1 Rehabilitación Virtual

La rehabilitación virtual puede ser definida como la utilización de recursos de realidad virtual (RV), para promover la rehabilitación motora. El término fue acogido en el 2002 por los profesores Daniel Thalmann y Grigore Burdea¹. Este tipo de rehabilitación se sustenta en el entrenamiento basado en ejercicios de simulación repetibles y seguros que están dentro de entornos virtuales controlables.

En [3], [5], [17] y [18], se evidencian los beneficios proporcionados por la realidad virtual en rehabilitación motriz. Los ejercicios realizados durante el proceso de rehabilitación motora, que son utilizados tradicionalmente en las clínicas, son repetitivos y poco atractivos para los pacientes. La falta de motivación del paciente impide que dé continuidad a su tratamiento. La realidad virtual posibilita la utilización de ambientes sintéticos interactivos que ofrecen ejercicios inmersivos con trasfondo lúdico, lo cual mejora la motivación de los pacientes, su esfuerzo y adherencia al tratamiento.

La realidad virtual posibilita que en la realización de las actividades, el sistema sea el que se adapta a la discapacidad de cada paciente y se personalice a través de configuración en el software, de modo que pacientes con distintos cuadros patológicos puedan disfrutar de las mismas aplicaciones. Esta capacidad de adaptación puede generar reducción en costos que representan los procesos de rehabilitación, lo que puede considerarse como una de las principales ventajas y estímulos para que la rehabilitación virtual sea empleada por instituciones de salud.

Durante los procesos de rehabilitación muchos de los equipos convencionales usados tienen un bajo nivel de precisión, de modo que los profesionales de la salud podrían proporcionar diagnósticos no adecuados, esto debido a la interpretación de las mediciones erróneas. La mayoría de las aplicaciones de realidad virtual proporcionan datos precisos que permiten valorar objetivamente el estado de los pacientes además de permitir hacer seguimientos al paciente para observar su evolución durante el proceso de rehabilitación.

Las instituciones de rehabilitación están concentradas en cascos urbanos, por lo que los pacientes deben desplazarse para realizar tratamientos. La realidad virtual permite que sistemas no presenciales de tratamiento que incluyan monitoreo a distancia sean desarrollados, con lo que la rehabilitación puede transferirse a la casa de los pacientes, aumentando así la comodidad y disminución de los costos para el mismo.

¹Virtual rehabilitation, website: http://en.wikipedia.org/wiki/Virtual_rehabilitation

1.3 Realidad virtual.

Existen múltiples definiciones de realidad virtual, que involucran distintos aspectos, una definición sintetizada puede ser:

“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” [2].

La realidad virtual en aplicaciones médicas tuvo su crecimiento motivado por la posibilidad de entrenamiento y familiarización con algunos casos clínicos por parte del personal médico, anteriormente estas situaciones eran practicadas en pacientes reales. Las ventajas que la realidad virtual trajo a la ciencia médica fueron revolucionarias, desde las primeras aplicaciones surgidas al inicio de los años 90, por la necesidad de visualización de datos médicos complejos, particularmente para asistir la planeación de cirugías o la misma cirugía. La primer área en explorar el poder de la realidad virtual en la medicina, fue la educación de médicos en entrenamiento de sus habilidades a través de simuladores quirúrgicos análogos a los simuladores de vuelo, pioneros en la utilización de la tecnología de realidad virtual. El objetivo de estos simuladores es principalmente probar la enseñanza cognitiva y las habilidades manuales de cirujanos y residentes; actualmente los simuladores quirúrgicos son utilizados en oftalmología, urología, ortopedia, cirugía neurológica, cirugía cardiaca, entre otras. Respecto a los usos de ambientes virtuales en terapia, estos se han popularizado como instrumentos de apoyo a aplicaciones para fines de rehabilitación física y control de actividades motoras [19].

Pese a la dificultad para precisar una definición respecto a RV, existe casi una generalización referente a las características principales de la realidad virtual. Estas se especifican en la Tabla 1.1.

| | |
|-----------------------|--|
| Inmersión | Hardware y software se combinan para dar una sensación de inmersión o presencia en el ambiente virtual. |
| Interactividad | El computador genera un ambiente que responde en tiempo real a los movimientos y acciones del usuario. |
| Lógica interna | El medio ambiente generado por el computador tiene su propio orden físico, moral y social, lo cual puede corresponder en menor o mayor proporción a lo que se presenta en el mundo real. |

Tabla 1.1. Características principales de los ambientes virtuales.

1.3.1 Aspectos generales de la Realidad Virtual

En términos generales la realidad virtual se basa en la aplicación de tres conceptos: participación, interacción e inmersión, esto para que el ambiente

sintético provoque respuestas cognitivas del usuario, propias de situaciones reales que están siendo simuladas por el computador; dichos conceptos se detallan a continuación.

Participación

El concepto de participación es la capacidad de estimular la motivación para que el usuario interactúe con el ambiente virtual. La reproducción de realidad en un ambiente sintético atrayente despierta en el usuario la curiosidad de simular sus actividades habituales en el escenario virtual.

La participación puede ser pasiva, como por ejemplo leer un libro o ver televisión, o activa, como en el caso de realizar una actividad en conjunto con otra persona. La realidad virtual tiene el potencial para trabajar los dos tipos de participación al permitir la exploración de un ambiente virtual y proporcionar la interacción dinámica del usuario con el mundo virtual.

Interacción

La interacción consiste en la capacidad que tiene el computador para detectar las acciones de usuario en el ambiente virtual y modificarlo mediante el curso de la simulación. Dicha característica determina la capacidad reactiva del ambiente y atrae a los usuarios por el hecho de que sus acciones se reflejan en el escenario virtual, además de eso, la facilidad de interacción con el usuario es el resultado de colocar en práctica sus conocimientos intuitivos de manipulación y navegación traídos del mundo real.

De acuerdo con el nivel de interacción podemos clasificar las experiencias de realidad virtual de tres maneras: pasiva, exploratoria o interactiva.

En 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.

A diferencia de la realidad pasiva, 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.

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, Acrópolis, 400 A.C Enciclopedia Microsoft Encarta 2009.

Inmersión

La idea de inmersión está ligada al hecho de despertar en el usuario alguna sensación de presencia en el ambiente. Un sistema virtual puede proporcionar varios niveles de inmersión dependiendo de los dispositivos de hardware empleados.

De acuerdo al concepto de inmersión se podrían clasificar la realidad virtual en: no Inmersa e Inmersa. Una forma intermedia, denominada realidad virtual semi-inmersa, utiliza recursos que promueven un grado limitado de inmersión por parte del usuario.

La realidad virtual no inmersa es una modalidad de realidad virtual que hace uso solamente de equipos comunes como: monitor, teclado y mouse. Por el hecho de utilizar recursos de fácil acceso.

Entre tanto la realidad virtual inmersa 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 (ver Figura 1.2); sin embargo, dispositivos

ligados a otros sentidos también son importantes para sentir la inmersión, como dispositivos de retorno de fuerza o sonidos.



Figura 1.2. Caverna virtual.

La facilidad de inmersión del usuario permiten que habilidades y conocimientos intuitivos del usuario puedan ser utilizados en la manipulación de objetos virtuales, y el objetivo es que el usuario tenga la impresión de estar actuando dentro del ambiente virtual, apuntando, golpeando, manipulando o ejecutando otras acciones sobre los objetos virtuales, en tiempo real.

1.3.2 Aspectos que consideran en diseño de una aplicación con ambientes virtuales

Durante el desarrollo de ambientes virtuales se debe considerar algunos requisitos para que el ambiente sea atractivo y responda a las expectativas del usuario. Claudio Kirner [20], indica que un ambiente virtual debería cumplir con algunos de los requisitos enunciados en la Tabla 1.2.

| | |
|-----------------------|--|
| 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 1.2. Aspectos considerados en Realidad Virtual.

1.4 Sistemas Hápticos

Usualmente las personas exploran objetos en varias etapas, primero hacen una inspección visual del ambiente con sus ojos, para encontrar la posición del objeto, después tocan el objeto para sentir su forma general, finalmente es hecha una exploración manual más cuidadosa para investigar las características de la superficie del material que forma el objeto como, su textura, su forma y consistencia. En el contexto computacional esta exploración concierne a la subárea de la realidad virtual llamada sistema háptico [21].

El término háptico se deriva de la palabra griega “hápticos” que significa “tocar o percibir” y relaciona la información sensorial recibida a través del tacto o contacto físico. Se sabe que el tacto es el único sentido humano bidireccional, lo que implica que recibe y envía información. Al tocar, una fuerza es aplicada sobre algo y su resultante es percibida por quien la aplicó.

Así, debido a las características del sistema sensorial táctil, realizar o percibir contacto depende necesariamente del contacto directo con el objeto tocado.

De acuerdo con Burdea e Coiffet [21], dos diferentes conceptos son normalmente mencionados en relación a la percepción del contacto: tanto el tacto como la cinestesia. El tacto está relacionado con el hecho de sentir diferentes tipos de sensaciones como temperatura, presión, vibración o dolor y depende de la sensibilidad cutánea, esta sensación varía de acuerdo con la región utilizada para realizar el contacto. Entre tanto la cinestesia (griego: kinesis “movimiento” y aisthesis: sensibilidad) se refiere a la percepción de las tensiones aplicadas a los músculos y articulaciones, esto permite tener capacidad de reconocer la localización espacial del organismo, sin utilizar la visión. Este tipo específico de la percepción permite el mantenimiento del equilibrio y la realización de diversas actividades, por ejemplo, caminar sin mirar dónde poner los pies.

En sistemas computacionales, la identificación del tacto y del retorno de fuerza depende de dos partes igualmente importantes que forman los sistemas hápticos: los dispositivos, responsables de recibir acciones del usuario y presentar las propiedades relacionadas al tacto, y las rutinas de control, responsables de calcular y enviar las propiedades del tacto al dispositivo.

En el ser humano, las sensaciones del tacto y retorno de fuerza no pueden ser separados, algo similar sucede con los dispositivos hápticos.

Los dispositivos hápticos son periféricos de entrada y salida que permiten al usuario interactuar en un ambiente virtual simulando la sensación correspondiente al tacto, en el cual retornan fuerzas o sensaciones referentes a

las características de los objetos. De esta forma, es posible definir los dispositivos hápticos como interfaces hombre-computador que asocian respuestas al tacto, con el propósito de proveer un medio de comunicación más natural entre personas y máquinas.

Una característica fundamental de esos dispositivos es la posibilidad de ser programados. Este concepto se refiere a la capacidad que los dispositivos hápticos pueden modificar sus propiedades mecánicas y físicas a través de rutinas provenientes del computador. Con eso, se crea la posibilidad de un intercambio bidireccional de información entre usuario y sistema.

En la Figura 1.3, se hace una comparación entre el sentido del tacto utilizando un dispositivo convencional, por ejemplo el mouse y este mismo sentido utilizando dispositivos hápticos.

Con el mouse convencional, el flujo de información referente al sentido del tacto tiene apenas una dirección, del usuario hacia el computador. De esta forma, el usuario solamente recibe información sobre sus movimientos de forma visual por lo que puede apreciar en la pantalla del computador.

Entre tanto, con el dispositivo háptico, cuando el usuario realiza un movimiento interactuado con el ambiente virtual, más allá de una retroalimentación visual, rutinas de control del ambiente responden enviando un vector de fuerza al dispositivo, que a su vez puede retornar al usuario una retroalimentación táctil, permitiendo una interacción rápida e intuitiva con el computador.

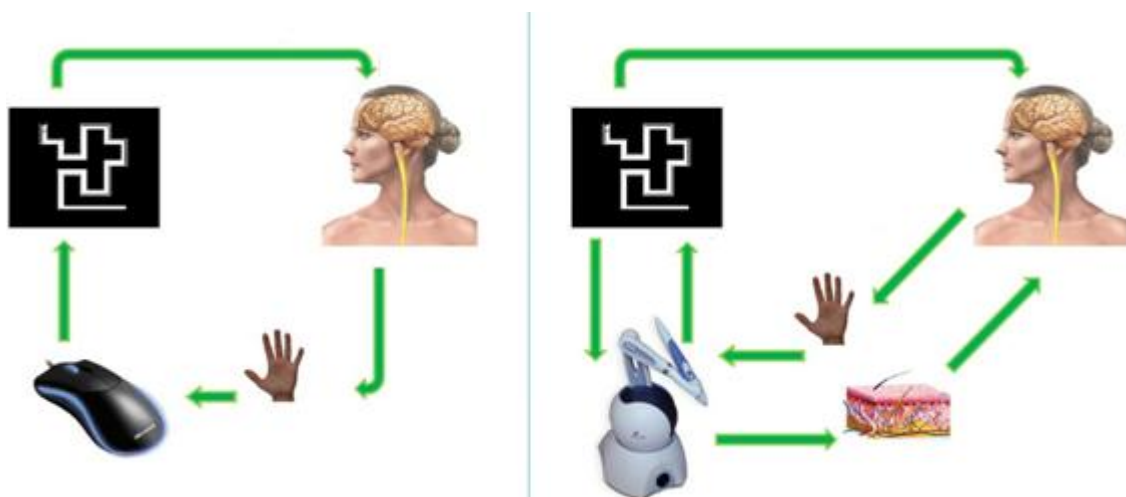


Figura 1.3. Comparación del flujo de información en una aplicación de RV empleando un dispositivo háptico y un dispositivo convencional (elaboración propia).

1.4.1 Modelado de la escena háptica.

El modelado de la escena háptica en ciertos aspectos es semejante al de una escena gráfica. Esta última está formada básicamente por un volumen de visualización gráfica dentro del mundo virtual y por los objetos virtuales dentro de ese volumen. De manera similar, la escena háptica también define el volumen dentro del mundo virtual, conocido como volumen háptico, y posiciona los objetos dentro de ese volumen, siendo uno de los objetos de esta escena el propio dispositivo háptico. En la escena háptica son irrelevantes las características visuales de los objetos, tales como color e iluminación. En este caso los objetos presentan propiedades, como aspereza, dureza y elasticidad que permitan identificarlos a través del tacto, de modo que sin necesidad de visualizar el objeto se puede percibir la interacción, sentir su textura, variaciones superficiales, etc.

Para que un ser humano tenga la impresión de movimiento continuo de un objeto en la pantalla además de que pueda ver las alteraciones del mismo, el contenido de esa escena gráfica debe ser refrescada entre 30 y 60 veces por segundo, a una frecuencia menor el usuario percibirá discontinuidades o parpadeos en la imagen, con lo que se perderá la sensación de realismo. Para detectar las características físicas de un objeto, paralela y sincronizadamente a una visualización gráfica, las rutinas de cálculo de la escena háptica necesitan ser ejecutadas a una frecuencia de 1000Hz [22], una vez modificada la posición del dispositivo en el ambiente virtual es enviada la señal de retorno de fuerza, que será percibida por el usuario, al cálculo de fuerzas que debe producir el dispositivo en cada ciclo se le llama renderización háptica.

1.4.2 Renderización Háptica

De acuerdo con Basdogan y Srinivasan [23], la renderización háptica puede ser definida como el proceso por el cual las rutinas de control háptico calculan las alteraciones en la escena háptica, actualizando y enviando las modificaciones al usuario en tiempo real.

El proceso de renderización háptica puede ser descrito de la siguiente manera: el usuario manipula el dispositivo háptico cuya representación visual puede ser vista en la escena gráfica, dicha representación tiene varias denominaciones entre las que están: HIP (Haptic interface point), punto de interacción, cursor o proxy. A medida que el usuario mueve el proxy a lo largo de la escena, su posición y orientación son detectadas por los encoders o sensores del dispositivo; cuando hay una colisión o interacción con el objeto virtual, ésta debe ser detectada e inmediatamente pasan a actuar algoritmos de cálculo de

retorno de fuerza. En la mayoría de aplicaciones cuando el proxy interactúa con los objetos, los algoritmos calculan la fuerza de relación basada en la profundidad de penetración. Los vectores de fuerza pueden entonces ser modificados de forma que pueden representar el tipo de superficie que se desea renderizar [23].

Las principales etapas de la renderización háptica son:

- Localización del proxy en el ambiente virtual.
- Detección de la interacción entre el proxy y la geometría de los objetos virtuales.
- Cálculo del vector de fuerza de reacción (puede basarse en las propiedades de los materiales definidas en el ambiente virtual).
- Envío del vector de fuerza al dispositivo háptico.

El renderizado háptico requiere ser ejecutado a una frecuencia de 1000 veces por segundo, para que el usuario pueda tener una sensación estable y coherente que garantice sensaciones de contacto con los objetos. A una frecuencia menor el dispositivo podría producir vibraciones.

1.4.3 Propiedades de los materiales

Los sistemas hápticos son capaces de simular propiedades de los materiales para distinguir los elementos de la escena [22]. Es posible simular varios materiales por medio de combinaciones de ciertos valores numéricos y fórmulas matemáticas.

Las propiedades de los materiales soportadas por el dispositivo Phantom son: stiffness, damping, static friction, dynamic friction y popthrough.

Stiffness, controla la rigidez de la superficie del objeto y determina la tasa con la que la fuerza aumenta cuando el proxy del dispositivo háptico intenta penetrar la superficie. La fuerza es regida por la Ley de Hook y es comúnmente usada en renderización háptica [18], su expresión está dada por (1.1), en la cual, el coeficiente k representa la rigidez (stiffness), y varía entre 0 a 1, x representa la profundidad de penetración en la superficie o vector de desplazamiento.

$$F = -k \cdot x \quad (1.1)$$

Como puede verse en la Figura 1.4, el resorte es colocado entre la posición fija P_0 y la posición del dispositivo háptico P_1 . La posición fija está usualmente situada en la superficie del objeto que el usuario está tocando con el dispositivo háptico. La profundidad de penetración ($x = P_0 - P_1$) es tal que la fuerza del resorte se encuentra siempre direccionada en la orientación de la posición fija. La fuerza sentida se llama fuerza de restauración del resorte, ya que la misma está intentando llevar al resorte hasta su estado de reposo, que en este caso es cero. Un valor bajo en la constante de elasticidad permitirá tener una sensación de suavidad, entre tanto un valor alto permitirá tener la sensación de rigidez.

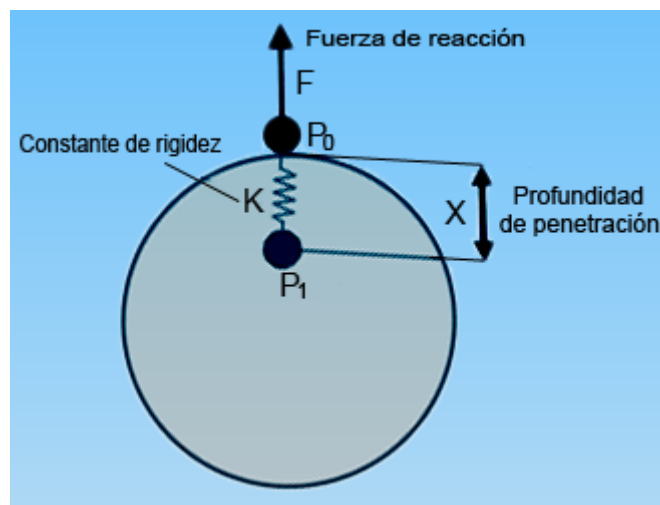


Figura 1.4. Fuerza de restauración, representada gráficamente.

La fuerza amortiguadora (damping), tiene como principal utilidad reducir la vibración de contacto en la superficie del objeto y es proporcional a la velocidad final del punto de interacción del dispositivo háptico. La fórmula empleada se muestra en (1.2), donde b es la constante de amortiguamiento y v es la velocidad del dispositivo.

$$F = -b \cdot v \quad (1.2)$$

Cuando el proxy o punto de interacción se mueve sobre la superficie de algún objeto virtual o del mismo espacio de trabajo háptico podría presentarse una resistencia al movimiento. Dicha resistencia se denomina fuerza de fricción.

Existen distintas formas de simular fricción con dispositivos hápticos, estas incluyen la fricción coulúmbica, fricción viscosa, fricción estática y la fricción dinámica.

La fricción de Coulomb, es una fuerza que tiene amplitud constante con respecto al cambio de velocidad, pero el signo de la fuerza de fricción cambia al invertir la dirección de la velocidad, como se ve en la Figura 1.5(c), no se tiene en cuenta la fuerza de fricción cuando la velocidad es cero, debido a su función signo; este modelo se expresa en (1.3); de donde μ representa la constante de fricción y \dot{x} la velocidad del efector final.

$$F = \mu \cdot \text{sgn}(\dot{x}) \quad (1.3)$$

$$\text{sgn}(\dot{x}) = \begin{cases} 1 & \text{si } (\dot{x}) > 0 \\ -1 & \text{si } (\dot{x}) < 0 \end{cases}$$

Usualmente se emplea la fricción de coulomb para ayudar a que las transiciones sean suaves cuando el proxy cambia de dirección.

La fricción viscosa representa una relación lineal entre la fuerza aplicada y la velocidad, usualmente la representan con un amortiguador. Su expresión se muestra en (1.4), de donde B es el coeficiente de fricción viscosa y v es la velocidad relativa entre dos superficies (ver Figura 1.5(a)).

$$F_v = B \cdot v \quad (1.4)$$

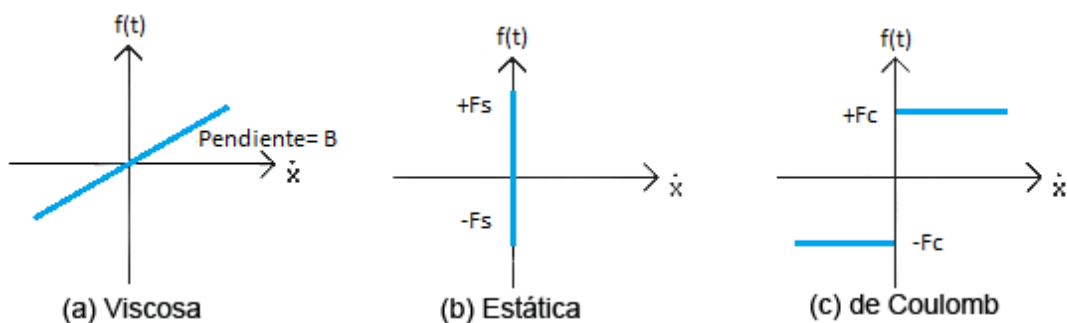


Figura 1.5. Tipos de fricción.

La fuerza de fricción estática representa una fuerza que tiende a impedir el movimiento desde el comienzo y desaparece una vez se inicia el movimiento. Esta fricción se representa por (1.5), en la cual, μ_e recibe el nombre de

coeficiente de fricción estática y n es la magnitud de la fuerza ejercida en por una superficie sobre otra (ver Figura 1.5(b)).

$$F_e = \mu_e \cdot n \quad (1.5)$$

En ambientes virtuales la fricción estática ofrecería resistencia al punto de interacción o proxy al inicio del movimiento del mismo. A diferencia de lo anterior la fuerza de fricción dinámica opone resistencia solamente durante el movimiento del proxy sobre las superficie del objeto o en el espacio de trabajo háptico.

Popthrough, o resistencia, representa la fuerza necesaria para atravesar un objeto. Un valor de cero significa que la superficie no puede ser atravesada.

Para el caso del dispositivo Omni Phantom las propiedades de los materiales se definen haciendo uso de las funciones mostradas en la Tabla 1.3.

| | |
|-------------------------------------|--|
| Stiffness | hlMaterialf(HL_FRONT_AND_BACK,HL_STIFFNESS, 0.7); |
| Damping | hlMaterialf(HL_FRONT_AND_BACK, HL_DAMPING, 0.1); |
| Fricción estática y dinámica | hlMaterialf(HL_FRONT_AND_BACK, HL_STATIC_FRICTION, 0.2); hlMaterialf(HL_FRONT_AND_BACK, HL_DYNAMIC_FRICTION,0.3); |
| Popthrough | hlMaterialf(HL_FRONT_AND_BACK, HL_POPTHROUGH, 0.5); |

Tabla 1.3. Funciones que llaman a propiedades de materiales usando dispositivo háptico Omni Phantom.

1.4.4 Principales dispositivos hápticos disponibles en el mercado

Son múltiples los dispositivos hápticos comerciales disponibles en la actualidad; por un lado están los dispositivos hápticos con interactividad limitada, tales como: los *gamespads* con vibración y los joysticks con retorno de fuerza, por otro lado están los dispositivos con mayor interactividad, pues pueden permitir mayores grados de libertad en movimientos de traslación, rotación y retorno de fuerza. De estos últimos dispositivos los más destacados son los guantes hápticos (*data gloves*) y los dispositivos con único punto de interacción (*point-data device*), estos son descritos a continuación.

Guantes Hápticos

Los guantes hápticos fueron desarrollados para reconocer los movimientos de los dedos de quien los porta; normalmente la información de dichos movimientos es enviada a ambientes virtuales.

Estos dispositivos están compuestos por un exoesqueleto que permite la captura de información, como la flexión de los dedos y la orientación real de la mano. El retorno de fuerza se ejerce por medio de una red de tendones, dispuestas para acoplar los dedos, por lo cual estos dispositivos poseen múltiples puntos de interacción. Con los guantes hápticos es posible tener la sensación de agarrar un objeto además de tocarlo. Es éste el caso del dispositivo Cybergrap de Immersion Corporation visto en la Figura 1.6 (b), que además de explorar las propiedades físicas de los objetos en ambientes virtuales también permite manipularlos.

Immersion Corporation también cuenta con el dispositivo CyberTouch que es un guante que cuenta con pequeños vibradores electromecánicos, dispuestos en cada dedo y en la palma de la mano, en conjunto pueden generar y variar las sensaciones táctiles simples, como pulsos o vibraciones sostenidas, los cuales pueden ser usados en combinación para producir patrones complejos de retroalimentación táctil, el CyberTouch visto en la Figura 1.6(a), es la versión preliminar al Cybergrap, puesto que permite explorar objetos de ambientes virtuales pero no está en capacidad de agarrar o sostener dichos objetos.



(a) CyberTouch

(b) Cybergrap

Figura 1.6. Guantes fabricados por Immersion Corporation.

Dispositivos con único punto de interacción

En los dispositivos con puntos de interacción limitados la sensación táctil o de fuerza es definida casi siempre en un único punto. Estos dispositivos son capaces de rastrear los movimientos del usuario en los ejes x , y , y z en ambientes virtuales tridimensionales, los motores al interior del dispositivo retornan las fuerzas pre-programadas cuando son detectadas las colisiones o interacciones con los objetos virtuales dentro del ambiente, simulando una sensación táctil. Entre los fabricantes que ofrecen dispositivos hápticos de este tipo, con mayor acogida en el mercado están los mostrados en la Tabla 1.4, en ésta, también se pueden observar las características de mayor relevancia de los mismos.

| Fabricante | Dispositivo | Espacio de Trabajo | | | | | | GDL | Fuerza (N) | | Torque (N.m) | |
|-------------------------|---|--------------------|-------|-------|----------------------|--------|----------|-----|------------|----------|--------------|-------------|
| | | Traslación (mm) | | | Rotación (°) | | | | Máximo | Continuo | Máximo | Continuo |
| | | x | y | z | φ | ψ | θ | | | | | |
| Haption SA | Virtuose™ 6D Desktop. Figura 1.7(a) | 120 (Esférico) | | | 35 Desde el centro. | | | 6 | 5-10 | 1.4-3 | 0.2 - 0.5 | 0.06 - 0.14 |
| | Virtuose™ 6D. Figura 1.7(b) | 450 (Cubico) | | | | | | | 35 | 10 | 3 | 1 |
| Novint Technologies Inc | Novint Falcon . Figura 1.7(c) | 101.6 | 101.6 | 101.6 | | | | | >8.9 | | | |
| | Pistol Greep. Figura 1.7(d) | | | | | | | | | | | |
| MPB Technologies | The Freedom 6S. Figura 1.7(e) | 170 | 220 | 330 | | | | 6 | 2.5 | 0.6 | | |
| | The Freedom 7S. Figura 1.7(f) | | | | | | 7 | | | | | |
| Sensable Technologies | PHANTOM Premium 1.5/6DOF. Figura 1.7(g) | 381 | 267 | 191 | 297 | 260 | 335 | 6 | 8.5 | 1.4 | 8.5 | 1.4 |
| | PHANTOM Omni Device. Figura 1.7(h) | 160 | 120 | 70 | | | | 6 | 3.3 | 0.88 | 3.3 | 0.88 |
| Forse Dimension | Omega.6. Figura 1.7(i) | ∅160x110 | | | 240 | 140 | 320 | 6 | 12 | | | |
| | Omega.7. Figura 1.7(j) | | | | 235 | 140 | 200 | | 20 | | | |
| | Delta.6. Figura 1.7(k) | ∅360x300 | | | 40 | | | | 20 | | 0.2 | |
| FCS Robotic | Haptic Master Figura 1.7(l) | 360 | 400 | 280 | 144 Desde el centro. | | | 3 | 250 | 100 | | |

Tabla 1.4. Principales fabricantes y características más importantes de dispositivos hápticos disponibles en el mercado.



(a) Haption SA, The Virtuose Desktop 6.



(b) Haption SA The, Virtuose 6D.



(c) Novint Technologies, Novint D Falcon.



(d) Novint Technologies, Pistol Greep.



(e) MPB Technologies, The Freedom 6S.



(f) MPB Technologies, The Freedom 7S.



(g) Sensable Technologies, Phantom Premium 1.5/6DOF.



(h) Sensable Technologies, Omni Phantom.



(i) Forse Dimension, Omega.6.



(j) Forse Dimension, Omega.7.



(k) Forse Dimension, Delta.6.



(l) FCS Robotics Haptic Master.

Figura 1.7. Principales dispositivos hápticos disponibles en el mercado.

1.5 Aplicación de realidad virtual en terapias de rehabilitación de motricidad fina.

En esta sección se muestran algunas aplicaciones de realidad virtual basada en dispositivos hápticos, con algunas características comunes a las del Ambiente Virtual para Rehabilitación de Motricidad Fina propuesto en este trabajo.

Para empezar se presenta el prototipo desarrollado por la escuela de fisioterapia del Trinity College Dublín en conjunto con el hospital St James, en Irlanda [6], estos crearon una aplicación con la que mostraron que la atención y la motivación estimuladas por entornos virtuales son dos factores importantes que inciden en el re-aprendizaje motriz después de haber sufrido una lesión. Se muestran estos aspectos sumados al uso de la tecnología háptica como un medio para llevar a cabo terapias en pacientes con problemas motrices en miembro superior. El prototipo del sistema se muestra en la Figura 1.8; básicamente está compuesto por un ambiente diseñado en tres dimensiones, una interfaz háptica, y una serie de elementos que permiten al paciente tener una postura adecuada y confortable durante la realización de los ejercicios.

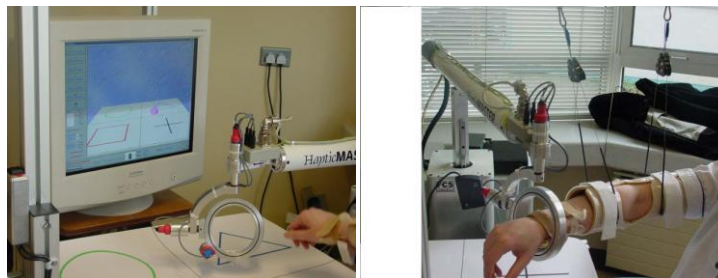


Figura 1.8. Prototipo del sistema GENTLES.

Esta aplicación está basada en el uso de la interfaz Haptic Master, consta de tres entornos virtuales en los que se realiza el entrenamiento; éstos se diferencian en la ambientación gráfica que poseen. El primer entorno es para la familiarización del usuario con el sistema. Los ejercicios realizados en el segundo entorno consisten en trayectorias rectas asistidas por una fuerza generada por la interfaz háptica de un punto a otro para completar un recorrido mayor fijado por el fisioterapeuta; los puntos del recorrido son cuatro y se encuentran ubicados en el plano. En el tercer entorno de entrenamiento se realiza un ejercicio que consiste en llevar un objeto de un lugar a otro sobre un plano representado por una mesa en el ambiente virtual; estos tres entornos de entrenamiento se denotan con las letras A, B y C respectivamente y pueden observarse en la Figura 1.9. En el artículo en el que se hace la publicación de

este trabajo, no se menciona si la fuerza se ejerce en el plano o en el espacio; pues sólo se habla de una fuerza de asistencia.

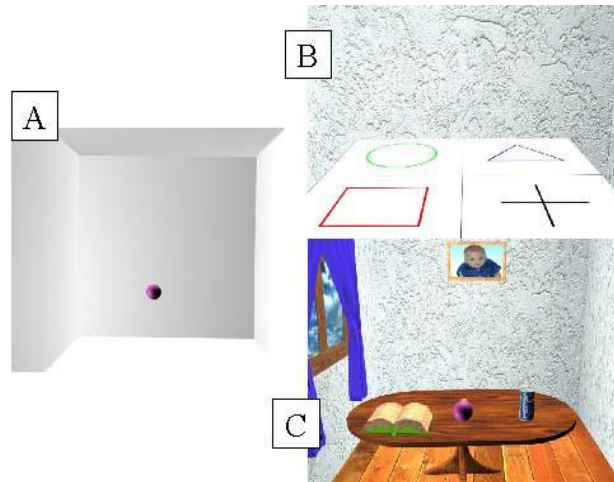


Figura 1.9. Entornos de entrenamiento en [6].

En el trabajo realizado por la Western Michigan University en conjunto con The University of South Florida [7], se crea un ambiente virtual en dos dimensiones para dibujar en él dos tipos de trayectorias: un laberinto y una trayectoria de forma irregular. La trayectoria que representa al laberinto se compone de tramos horizontales y verticales, por lo tanto el cambio de dirección se hace en ángulos rectos; en estos tramos de trayectoria se aplica de forma particularizada a cada tramo, una fuerza en el plano xy para guiar el recorrido; el dispositivo háptico usado es la interfaz Omni Phantom. En el otro tipo de trayectoria como es más compleja de modelar, la asistencia del dispositivo háptico se limita a la configuración de viscosidad y de inercia como forma de ayuda al desempeño del seguimiento de la trayectoria. Estos dos tipos de trayectorias implementadas en el ambiente virtual bidimensional se muestran en la Figura 1.10.

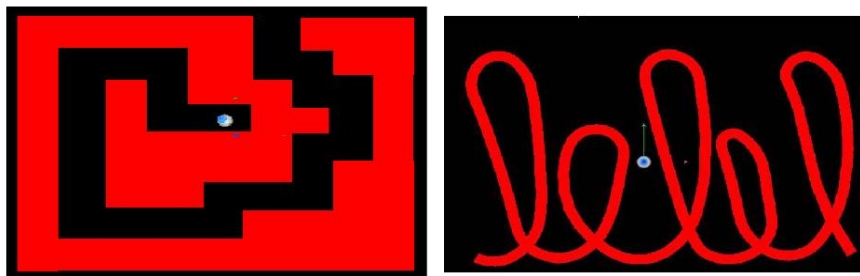


Figura 1.10. Trayectorias del ambiente bidimensional descrito en [7].

Una aplicación para realizar la medición de las habilidades motrices de miembro superior de pacientes que sufren enfermedades neurológicas, fue desarrollada mediante la implementación de un ambiente virtual en tres dimensiones, al que se conecta una interfaz háptica Phantom Premium [8]. La aplicación consiste en el seguimiento de un laberinto creado en el espacio, en el cual se puede sentir la realimentación de la fuerza cuando el cursor, que es representado por una esfera, colisiona con las paredes del mismo. Se realizan las mediciones de la fuerza del contacto, la velocidad del cursor, el número de colisiones, la frecuencia máxima del temblor en el miembro superior del paciente, el tiempo que dura la colisión, entre otras. En este entorno virtual no se provee un campo de fuerza de asistencia al seguimiento de la trayectoria, la realimentación de fuerza que se implementa es estrictamente para que el paciente perciba las colisiones con las paredes del laberinto. En la Figura 1.11, se muestra uno de los laberintos utilizados en esta aplicación.

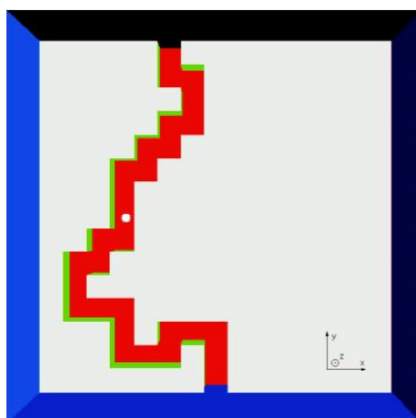


Figura 1.11. Laberinto utilizado para medición de habilidades motrices en [8].

El sistema que se detalla en [9], fue desarrollado en la University of New Brunswick, Canadá. El esquema de este sistema se puede observar en la Figura 1.12. El fin que se busca es el de medir en diferentes condiciones, el seguimiento que hace la mirada del paciente al orientarse en el ambiente virtual para tocar los objetos que están dibujados en él, es decir, obtener una medición que se relacione con la coordinación mano-ojo del paciente. En este caso la interfaz Phantom, se utiliza para que provea fuerzas de realimentación consecuentes con las colisiones de los objetos.

La forma en la que se ha adecuado el espacio de trabajo en este sistema, permite una inmersión mayor en el ambiente virtual por parte del paciente que realiza la práctica, ya que se hace coincidir el espacio de trabajo de la interfaz háptica, con los gráficos representados por el ambiente virtual en un mismo volumen de visualización que el paciente observa mediante el uso de unas

gafas estereoscópicas que tienen la función de hacer percibir profundidad y que además contienen el sensor que mide la orientación de la visión para después determinar el grado de coordinación mano-ojo.

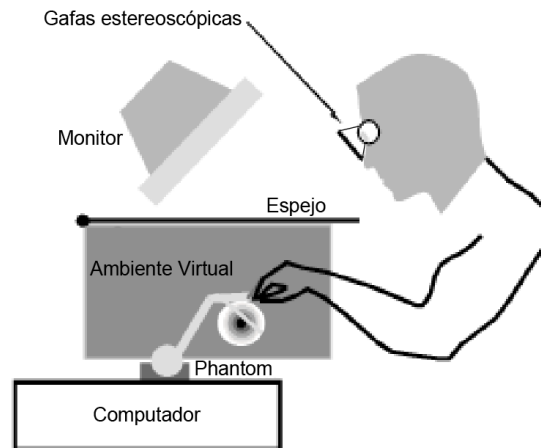


Figura 1.12. Esquema del sistema detallado en [9].

Capítulo 2

Herramientas de software y dispositivo háptico empleado

El avance y consolidación de las herramientas computacionales han facilitado el desarrollo de aplicaciones de software con propósitos de rehabilitación motriz en por lo menos dos aspectos: en el almacenamiento y administración de datos; y en 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.

El interés del presente trabajo radica en construir una aplicación de software con interfaz de usuario, que permita la administración de datos (paciente, cuadro patológico, desempeño, etc.) y que además cuente con un ambiente virtual con incorporación de dispositivo háptico que permita realizar ejercicios para rehabilitación de la motricidad fina. En este capítulo se muestra de forma general las herramientas de software que forman el compendio de aplicaciones que permitieron llegar al prototipo descrito en el Capítulo 4.

2.1 Selección de herramientas Software

Las herramientas de software empleadas para el diseño del prototipo descrito en el Capítulo 4, son en su mayoría de código abierto, y prestan facilidades para su programación. Todo el desarrollo de la aplicación se realizó en el lenguaje de programación C++, y se trabajó sobre la plataforma de Visual Studio 2008. Para la implementación de la interfaz de usuario se empleó Qt Creator, para el almacenamiento y administración de datos se utilizó SQLite. En la creación del ambiente virtual e interacción intuitiva con el mismo fueron

requeridas las librerías de OpenGL y las APIs del dispositivo háptico Omni Phantom (Open Haptics Toolkit, Version 3.0).

Algunas de estas herramientas ya han sido abordadas en distintos trabajos realizados dentro del programa de Ingeniería en Automática Industrial [24] y [25] (FIET, Universidad del Cauca), por lo que es prioridad en este capítulo profundizar en la exploración de la herramienta Open Haptics Toolkit, propia del dispositivo Phantom, ya que éste no ha sido incluido en ningún trabajo de grado de pregrado. La presentación de herramientas complementarias empleadas en el diseño del prototipo se exponen en el Anexo A de este documento.

2.2 Dispositivo Omni Phantom

El dispositivo háptico Omni Phantom, visto en la Figura 2.1, está compuesto por una base giratoria, que está unida a un grupo de brazos mecánicos en cuya extremidad se integra el efector final que se asemeja a un lápiz, éste, permite de modo natural simular diversos instrumentos reales como herramientas quirúrgicas, lápices, atornilladores, entre otros.



Figura 2.1. Dispositivo háptico Omni Phantom.

El dispositivo cuenta con 6 grados de libertad, los cuales tienen limitaciones físicas. Al tener 6 grados de libertad se permite el desplazamiento en los tres ejes del plano cartesiano y las tres rotaciones sobre los mismos ejes, permitiendo una retroalimentación de fuerza nominal de hasta 3.3 N, en los tres ejes.

En la Figura 2.2, se muestran las articulaciones con y sin retroalimentación de fuerza que presenta el dispositivo.



(a) Articulaciones motorizadas

(b) Articulaciones no motorizadas

Figura 2.2. Indicación de articulaciones del dispositivo háptico Omni Phantom.

De manera resumida se presentan en la Tabla 2.1 las características del dispositivo Phantom.

| Especificaciones técnicas | |
|---|---|
| Espacio de trabajo con realimentación de fuerza | 160 mm de ancho 120 mm de largo 70 mm de diámetro |
| Fricción | < 0.26 N |
| Fuerza Máxima Aplicable | 3.3N |
| Fuerza Continua Aplicable | >0.88N |
| Rigidez | Eje x 1.26 N/mm Eje y 2.31 N/mm Eje z 1.02 N/mm |
| Inercia | Aprox. 45 gr. |
| Peso | Aprox. 1.47Kg |
| Temperatura de Operación | De 10° – 35° |
| Temperatura de Almacenamiento | De -40° - 65° |
| Movimiento | 6 GDL |
| Feedback de Fuerza | X,Y,Z |
| Interfaz | IEEE-1394 Fire Wire, port: 6pin-6pin |
| Plataformas Soportadas | Intel ó PCs basados en AMD |
| Compatibilidad con OpenHaptics Toolkit SDK | OpenHaptics para Windows, Linux y Mac OS X |

Tabla 2.1. Especificaciones técnicas del dispositivo Omni Phamton.

2.3 OpenHaptics Toolkit

El OpenHaptics Toolkit es una biblioteca desarrollada por la empresa Sensable Technologies Inc, con el propósito de facilitar la incorporación, el uso y programación de los dispositivos hápticos en distintas aplicaciones, la versión 3.0, presenta una arquitectura dividida en 4 capas principales: HDAPI (HapticDevice API), HLAPI (Haptic Library API), QuickHaptics micro API, los driver del dispositivo Phantom (PDD), además de las capas mencionadas el Toolkit también cuenta con la librería Utilities y varios ejemplos con códigos fuente incluidos; la arquitectura mencionada puede verse en la Figura 2.3.

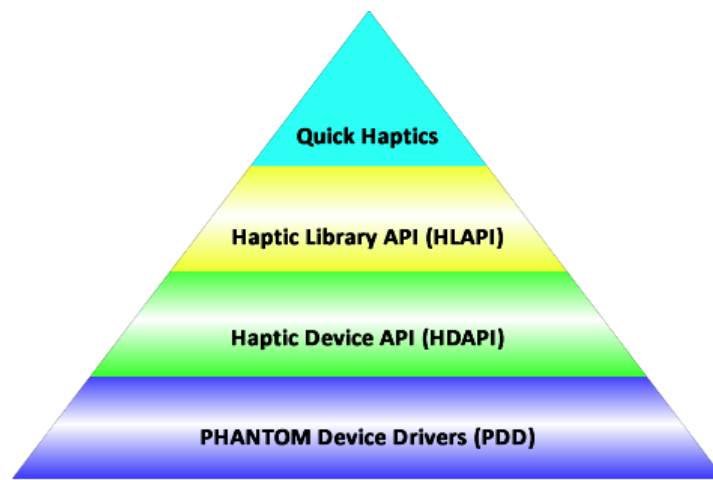


Figura 2.3. Arquitectura da biblioteca OpenHaptics Toolkit².

2.3.1 Phantom Device Drivers (PDD)

La instalación de los controladores del dispositivo Phantom son requeridos, para que este funcione correctamente. Pueden descargarse desde la página del fabricante, usando el serial del dispositivo.

2.3.2 HDAPI

La HDAPI (Haptic Device API – API del dispositivo háptico) proporciona funciones de bajo nivel para acceso y control directo del dispositivo, permite controlar la configuración del dispositivo durante la ejecución, además de la renderización de las fuerzas. En la Figura 2.4, se muestra la estructura de esta API.

La HDAPI, requiere que el desarrollador maneje directamente la renderización de la fuerza para el dispositivo háptico; para proporcionar dicha renderización

² Website: <http://www.sensable.com/products-openhaptics-toolkit.htm>

son requeridos algoritmos eficientes para el manejo de la estructura de datos, esto debido a la alta necesidad de actualización para estabilizar el lazo de control del dispositivo háptico.

Principales funciones de la HDAPI

Entre las principales funciones de esta API [22], están:

- Permitir el monitoreo del estado del dispositivo, referido a la posición y velocidad del mismo, además del monitoreo de las tres coordenadas en el espacio cartesiano y de tres de las articulaciones con la que cuenta.
- Permite la extracción de valores de velocidad y temperatura actuales del dispositivo, también del valor máximo de fuerza que el dispositivo está en capacidad de proporcionar.
- Permitir imponer al dispositivo fuerzas y torques en el espacio cartesiano y valores de voltajes a los motores
- Permitir la habilitación y deshabilitación de las fuerza de salida, además de la notificación de errores.
- Permite la calibración (reset de los encoders) automática o manual del dispositivo.
- Permite definir el uso de Callbacks Asíncronas y Síncronas
- Facilitar la sincronización de los hilos gráfico y háptico
- Definir la frecuencia del servo-loop (hilo háptico)

En la Tabla 2.2, se muestran las funciones principales de la API HDAPI.

| Algunas Funciones de HDAPI | |
|--|--|
| HdInitDevice (HD_DEFAULT_DEVICE) | Inicialización del dispositivo háptico. |
| HdEnable (HD_FORCE_OUTPUT) | Habilitación de las fuerzas. |
| hdStartScheduler() | Iniciar Scheduler. |
| HdUpdateCalibration (HD_CALIBRATION_ENCODER_RESET) | Calibración del Dispositivo (<i>reset dos encoders</i>). |
| HDCallbackCode HDCALLBACK DevicePositionCallback(void *pUserData) | <i>Callback</i> que retorna las coordenadas de posición (x,y,z) del dispositivo háptico. |
| HDCallbackCode HDCALLBACK DeviceVelocityCallback(void *pUserData) | Callback que retorna los valores de velocidad (x,y,z) del dispositivo. |
| HDCallbackCode HDCALLBACK DeviceForceCallback(void *pUserData) | <i>Callback</i> que impone los valores de fuerza al dispositivo. |
| hdGetDoublev(HD_CURRENT_POSITION, pPosition) | Función que permite obtener los valores de posición (x,y,z) del dispositivo. |
| hdGetFloatv(HD_CURRENT_VELOCITY, pVelocity) | Función que permite obtener El valor de la velocidad (x,y,z) del dispositivo. |

| | |
|--|--|
| hdSetDoublev(HD_CURRENT_FORCE, pForce) | Función que permite imponer los valores de fuerza al dispositivo háptico. |
| hdStopScheduler() | Función que permite parar la ejecución, de forma que sale del ciclo de servo-loop. |
| hdDisableDevice(hHD) | Función que permite desactivar El dispositivo háptico. |

Tabla 2.2. Funciones principales de la API HDAPI.

2.3.3 HLAPI

La HLAPI proporciona acceso al dispositivo a un mayor nivel de programación, pues fue desarrollada sobre la librería HDAPI, está basada en los conceptos de desarrollo de gráficos en tres dimensiones específicamente para ser compatible con OpenGL, facilitando así la sincronización del hilo háptico y gráfico. La estructura de esta librería puede apreciarse en la Figura 2.5.

Uso de OpenGL

El mecanismo primario de HLAPI usa para especificar la geometría de las figuras los comandos de OpenGL, estos podrían ser usados por la API e incluyen primitivas como puntos, líneas y polígonos así como arreglos simples de vértices, estos comandos se especifican usando *glBegin*. La captura de la geometría de OpenGL puede realizarse de dos formas diferentes, por el búfer de profundidad de la figura o el búfer de retroalimentación de la figura.

Existen varias bibliotecas externas construidas sobre OpenGL que permiten adicionar características no disponibles de manera directa en el propio OpenGL, entre estas están GLU y GLUT, la primera ofrece funciones de dibujo de alto nivel basadas en primitivas de OpenGL, la segunda, GLUT, facilita el manejo de ventanas e interacción con el mouse y teclado. HLAPI hace uso de dicha funcionalidad, por ejemplo, realiza transformaciones geométricas invocando funciones de OpenGL como *glTranslate*, *glRotate* y *glScale*, HLAPI aplica las transformaciones, enviadas por estos comandos a las primitivas geométricas de dibujo háptico. Además de las transformaciones geométricas también se pueden especificar geometrías predefinidas en GLU, como esferas, cilindros, discos.

Entre las principales prestaciones de la API HLAPI están [22]:

- Renderizado de fuerzas dependientes del movimiento, constante, modelo masa resorte, amortiguador, fricción etc.
- Especificación de las propiedades físicas sobre los materiales superficiales: fricción, dureza, amortiguamiento, sensibilidad por todas las caras etc.

- Definición y atención a eventos complementarios a la aplicación, como por ejemplo detección de la activación de los botones del efector final del dispositivo.

En la Tabla 2.3, se muestran las funciones principales de la API HLAPI.

| Algunas Funciones de HLAPI | |
|---|--|
| hHLRC = hCreateContext(hHD) | Crea un contexto para inicializar el dispositivo |
| hMakeCurrent(hHLRC) | Activa el contexto. |
| hBeginFrame | Marca el inicio de un marco háptico. |
| hEndFrame | Marca la finalización del marco háptico. |
| hBeginShape / hEndShape | Permite la especificación de las primitivas geométricas que serán enviadas al renderizado háptico. |
| hGenShape | Genera un identificador de figuras que puede ser usado con hBeginShape(). |
| HL SHAPE DEPTH BUFFER HL SHAPE FEEDBACK BUFFER | Capturan la geometría definida por los comandos de OpenGL, al usar HL SHAPE DEPTH BUFFER, solo son palpables las porciones de geometrías que sean visibles en las escena, entre tanto con HL SHAPE FEEDBACK BUFFER, se pueden tocar todas las caras. |
| hMatrixMode(HL_VIEWTOUCH_MATRIX) | Indica que se usara la matriz VIEWTOUCH, que permite mapear el espacio de trabajo a coordenadas visuales. |
| HL_PROXY_TRANSFORM | Especifica la transformación de la relación del proxy con las coordenadas del mundo. |
| shapeld | Identificador de la figura para la cual será detectado el evento. Se puede utilizar ANY_OBJECT para que el evento se detecte con cualquier figura háptica. |
| HL_CLIENT_THREAD | Indica que un hilo del cliente constatará los eventos que han ocurrido por medio de la función <i>hCheckEvents</i> . |
| HL_EVENT_TOUCH | Indica que se tocó una figura. |
| HL_EVENT_2BUTTONUP | Indica que se liberó el botón 1 del dispositivo. |
| HL_EVENT_UNTOUCH | Indica que se dejó de tocar la figura. |
| HL_EVENT_MOTION | Indica el movimiento del dispositivo. |
| HL_EVENT_1BUTTONDOWN | Indica que se presionó el botón 1 del dispositivo. |
| HL_EVENT_1BUTTONUP | Indica que se liberó el botón 1 del dispositivo. |
| HL_EVENT_2BUTTONDOWN | Indica que se presionó el botón 2 del dispositivo. |

Tabla 2.3. Funciones principales de la API HLAPI.

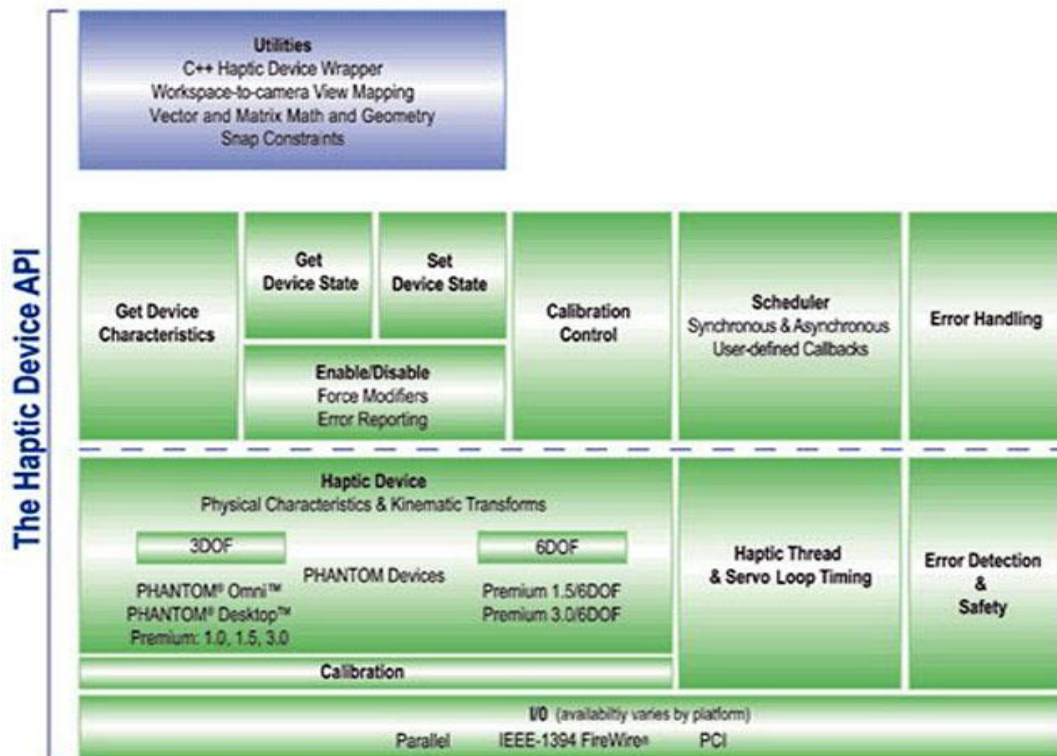


Figura 2.4. Funcionalidades de la librería HDAPI³.

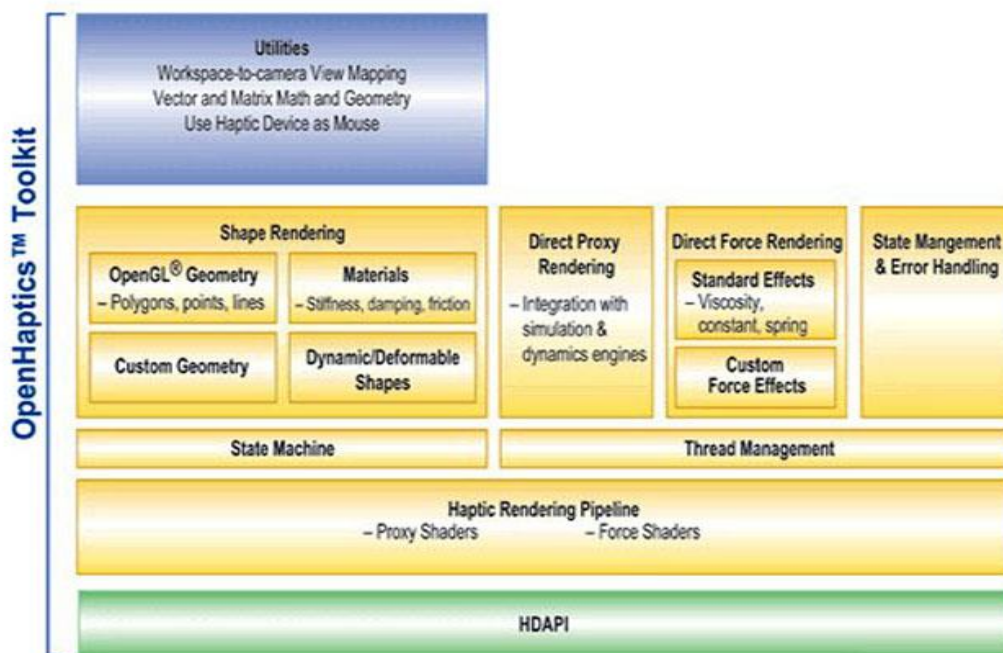


Figura 2.5. Funcionalidades de la librería HLAPI⁴.

³ <http://www.sensable.com/openhaptics-toolkit-hdapi.htm>

⁴ <http://www.sensable.com/openhaptics-toolkit-hlapi.htm>

HDAPI Versus HLAPI

La HDAPI requiere que el desarrollador gestione directamente la renderización de la fuerza para el dispositivo háptico mientras que HLAPI maneja el cálculo de renderizado háptico basado en primitivas geométricas, transformaciones y propiedades de los materiales. El renderizado realizado directamente por la HDAPI requiere que los algoritmos y las estructuras de datos de detección de colisiones y de renderizado de fuerza sean muy eficientes. Esto debido a la alta frecuencia con la cual la fuerza de retroalimentación del dispositivo requiere actualizarse. A diferencia la HLAPI evita que el programador deba implementar dichos algoritmos además de evitar la gestión de la sincronización de los lazos de control, las estructuras de datos y de estados.

2.3.4 Micro API QuickHaptics

La Micro API QuickHaptics permite que la creación de aplicaciones hápticas sea rápida y fácil, está construida encima de la HLAPI e implementada en C++. Esta API dispone de cuatro clases principales:

- DeviceSpace, permite manipular el espacio de trabajo a través del cual el dispositivo háptico puede moverse.
- QHRenderer, clase base para QHWin32 y QHGLUT. Renderiza las figuras de acuerdo a la vista de la cámara y permite realizar el renderizado de fuerza.
- Shape, esta clase soporta uno o más objetos geométricos, cuya representación es gráfica y háptica.
- Cursor, permite la representación gráfica del punto de interacción o proxy de la interfaz.

2.4 Desarrollo de una aplicación háptica usando el dispositivo Omni Phantom.

Los entornos virtuales como el que se desarrolla en este trabajo, se componen principalmente por un computador y uno o varios dispositivos hápticos, junto con otros dispositivos auxiliares que dependerán del tipo de aplicación. Para su funcionamiento requieren de dos escenas, gráfica y háptica que se hacen coincidir para que la representación cobre realismo.

La escena gráfica recoge información generada por la cámara previamente ubicada en el entorno virtual; compara la posición relativa de los objetos para determinar si hay superposición entre ellos y actualiza la imagen mediante los datos que tiene sobre la escena representada.

La información para la escena háptica proviene de la lectura del estado del dispositivo háptico, con estos datos se calculan la posible existencia de colisiones entre objetos cargados sobre esta escena, se determina la fuerza que debe aplicar cada motor y se reestructura la escena teniendo en cuenta desplazamientos y rotaciones. La escena háptica no conoce lo que sucede en la escena gráfica, solamente tiene conocimiento de los objetos 3D almacenados en su memoria; la cuestión es que estos objetos coincidan con los de la escena gráfica.

En la Figura 2.6, se muestra un esquema que representa las partes que componen una aplicación háptica, empleando OpenGL como motor de renderizado gráfico y las librerías de programación del dispositivo Omni Phantom para la representación de la escena háptica.

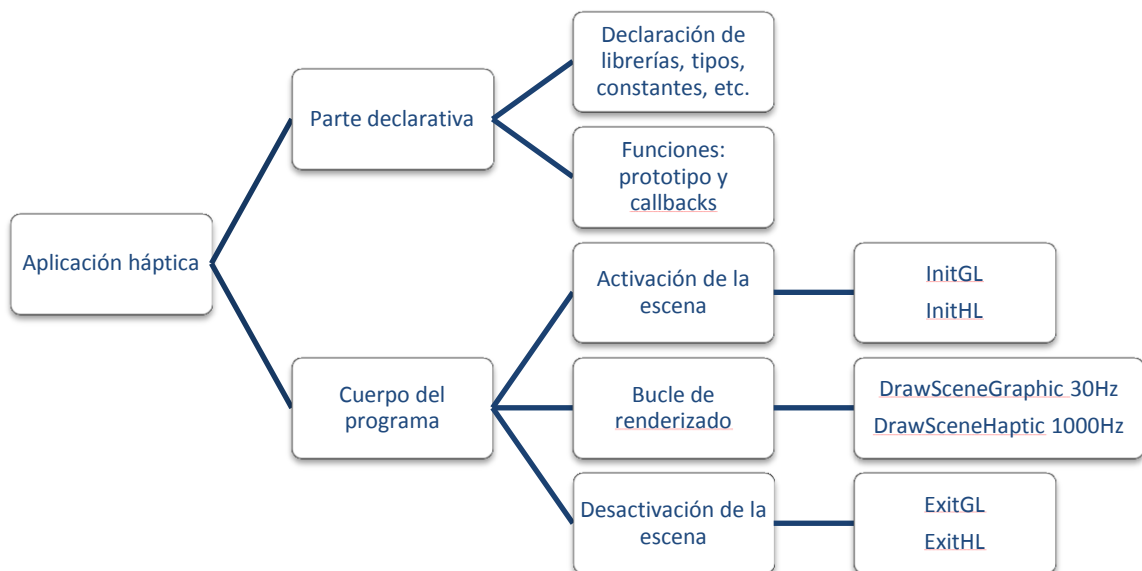


Figura 2.6. Estructura de una aplicación háptica.

Parte declarativa

En esta sección, como en muchas aplicaciones, se encuentran todas las inclusiones de librerías, declaración de variables globales, constantes, estructuras; que se determinen necesarias para el desarrollo de la aplicación. Igualmente es aquí donde se declaran las funciones y callbacks, las cuales son funciones específicas para el funcionamiento del dispositivo Omni Phantom y son llamadas cada vez que ocurre un evento.

Para poder incluir al dispositivo háptico en el entorno virtual, se necesita declarar un identificador del mismo; sin esto, no es posible leer su estado o enviarle órdenes de generación de fuerza; de la misma manera es necesaria la declaración de una variable conocida como contexto háptico, y tiene la finalidad de hacer que el dispositivo háptico al que se le ha dado un identificador quede inmerso en un entorno seguro en la aplicación. Las siguientes dos instrucciones son las que permiten la declaración del identificador del dispositivo y la declaración del contexto háptico:

```
static HHD ghHD = HD_INVALID_HANDLE;  
static HHLRC ghHLRC = 0;
```

Los objetos que se definen como palpables en la escena háptica también llevan su identificador que se declara en esta parte de la aplicación. Para el desarrollo del ambiente virtual que se describe en este trabajo en particular, puede omitirse esta declaración debido a que los modelos 3D que se cargan al ambiente virtual no son palpables de la forma predeterminada de la librería HL. La forma de declarar el indicador es como se muestra a continuación:

```
HLuint gFormaShapeId;
```

Cuerpo del programa

Después de la parte declarativa, está el cuerpo del programa que contiene el desarrollo de la aplicación comprendida en tres etapas principales:

- *Activación de la escena*
- *Bucle de renderizado*
- *Desactivación de la escena*

En la etapa de activación de la escena se deben inicializar tanto el entorno gráfico como el entorno háptico; en este caso se deben configurar aspectos de la librería OpenGL y HL respectivamente.

Activación de las escenas gráfica y háptica

Para configurar la escena gráfica, se crea una función denominada *initGL*, en donde se establece la forma en la que se va a presentar el entorno virtual; en esta función se eligen las opciones de OpenGL que serán aplicadas al ambiente como lo correspondiente a la iluminación de la escena donde se habilitan o deshabilitan las fuentes de luz, se les da una posición, y se escoge el tipo de luz que irradiarán, como luz focal, luz especular o luz de ambiente; la forma en la que se almacenan los estados de la escena capturados por la cámara, el aspecto que tendrán los objetos 3D cargados a la escena que se puede elegir entre suavizado de la superficie o sin suavizado para mirar las primitivas que forman el objeto; la manera en la que se tratan las superposiciones entre figuras, etc.

La escena háptica se configura con las opciones de la librería HL, lo cual se realiza en la función llamada *initHL*. En esta función lo primero que se hace es establecer una condición de inicio de la escena háptica, en el caso de que la interfaz háptica Omni Phantom no se encuentre conectada, se genera un mensaje de error y se termina la ejecución de la aplicación. Esta es una función muy importante del cuerpo del programa de cualquier aplicación desarrollada para trabajar con el dispositivo Omni Phantom, pues es aquí donde se realiza la introducción del dispositivo háptico en el contexto de la aplicación y se generan los identificadores que se asignan a los objetos 3D que van a ser palpables en el ambiente virtual. Además se define las partes de los objetos 3D que se tocarán, por ejemplo se puede elegir entre que sea palpable solamente la cara frontal o si también será palpable las partes que permanecen ocultas a la cámara.

Renderizado gráfico y háptico

Después de la inicialización de las escenas gráfica y háptica, se entra en el bucle de renderizado. Para realizar la renderización se definieron dos funciones llamadas *DrawSceneGraphic* para el renderizado de la parte gráfica, y *DrawSceneHaptic* para el renderizado háptico. En la función de renderizado gráfico, primero que todo se restablece el color de fondo, si esto no se hiciera, los objetos que se mueven en el entorno virtual como el cursor háptico no se borrarían y saturaría la pantalla de múltiples figuras del mismo. En esta sección se ubican todos los objetos 3D, 2D e incluso líneas, que se van a dibujar. Como el cursor háptico tiene una representación gráfica en el entorno virtual, la función que lo dibuja también se ubica dentro de esta función de renderizado gráfico.

En la función *DrawSceneHaptic* se ubican los objetos que serán palpables con el dispositivo háptico. Se necesita que estos objetos 3D se encuentren dentro

de las sentencias *hlBeginShape* y *hlEndShape* que los introducen en la escena háptica:

```
hlBeginShape (HL_SHAPE_FEEDBACK_BUFFER, gFormaShapeId);  
Objeto3D();  
hlEndShape();
```

El objeto 3D se identifica mediante *gFormaShapeId* que fue declarada como variable global y definida en la función de inicialización de la escena háptica; establece la forma en que la librería HL captura la geometría del objeto construido en la librería OpenGL con HL_SHAPE_FEEDBACK_BUFFER.

La escena gráfica y la escena háptica, utilizan generalmente las mismas figuras para que una escena sea consecuente con la otra; por lo tanto es recomendable que el dibujado o importado de figuras mediante OpenGL se realicen en una función aparte, de esta manera se podrá incluir en ambas escenas la misma figura solamente con el llamado a una función de dibujado.

Esta declaración del objeto como palpable se encuentra a su vez dentro de unas sentencias que definen el inicio y el fin del entorno háptico:

```
hlBeginFrame();  
hlEndFrame();
```

Dentro del *Frame*, además de los objetos palpables, se determina la forma en que las fuerzas serán generadas por el dispositivo, dependiendo del material que representan los objetos 3D cargados en la escena háptica, esto es el establecimiento de los valores de los diferentes coeficientes que caracterizan un material; como son los coeficientes de rigidez, amortiguamiento, fricción dinámica y fricción estática.

El renderizado háptico de la aplicación que se desarrolla en este trabajo, se realiza de una manera un poco diferente; como se necesita la implementación de un campo de fuerza, ya no se trabaja con la forma predeterminada de generación de fuerzas de la librería HL, sino que la fuerza se configura mediante una función callback en la que utilizando comandos de la librería HD (de más bajo nivel que la HL), se personaliza la fuerza que generará el dispositivo háptico. Dentro de este callback, el establecimiento de la fuerza se realiza dentro de las sentencias *hlBeginFrame* y *hlEndFrame*, las cuales tienen como argumento el identificador del dispositivo háptico.

Desactivación de las escenas gráfica y háptica

Los programas gráficos y de visualización tridimensional consumen una gran cantidad de recursos del computador, por lo que es conveniente indicar al procesador que debe liberar dichos recursos para que estén disponibles para otras aplicaciones. La desactivación de la escena gráfica permite dejar al sistema en un estado consistente que garantiza que la ejecución de otros programas no se va a ver afectada cuando utilicen los recursos empleados por la aplicación gráfica que acaba de cerrarse.

La escena háptica también debe desactivarse adecuadamente, porque si no se hace, queda guardada la información de la aplicación que se ha cerrado y cuando se ejecuta nuevamente se cargan datos basura dejados por la última ejecución. Lo que se debe liberar de la escena háptica son los identificadores de las figuras, el identificador del dispositivo y el contexto en el que está inmerso el dispositivo.

Información más detallada acerca de la creación de las escenas hápticas se puede consultar en [22].

Capítulo 3

Trayectorias y campo de fuerza

En el tratamiento de problemas relacionados con la motricidad fina de miembro superior, los ejercicios de entrenamiento que se realizan, incluyen el seguimiento de trayectorias de diferente tipo. Estas trayectorias pueden estar dibujadas sobre papel, como laberintos para ser seguidos mediante el uso de un lápiz; o bien pueden estar construidas en un material como plástico, madera o metal, para ser recorridas con un objeto que se sujeta con la mano.

Una de las prácticas más comunes consiste en el seguimiento de una trayectoria recta elaborada de un material conductor, inscrita en un pequeño aro metálico que la recorre por efecto del movimiento de la mano del paciente. El objetivo es sostener y desplazar el aro sin que éste haga contacto con la trayectoria; porque si esto ocurre, se cierra un circuito que enciende una luz indicando que el ejercicio no se realizó correctamente.

En el Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior desarrollado en el presente trabajo, se recrea este tipo de ejercicio de seguimiento de trayectorias, con el valor adicional de la implementación de un campo de fuerza para mantener el centro del anillo alineado con el eje de la trayectoria. El nivel de fuerza de dicho campo puede configurarse para obtener un mayor o un menor grado de asistencia en el momento en que el paciente se encuentra realizando el entrenamiento en el ambiente virtual.

Como se cuenta con la asistencia del campo de fuerza, al paciente le será más fácil realizar el ejercicio, por lo tanto la trayectoria no será una recta como en el ejercicio real, sino que en el ejercicio virtual se trata de trayectorias curvas de diferente apariencia.

Para que el entrenamiento en el ambiente virtual no resulte extremadamente difícil o que se presenten situaciones en las que el seguimiento de las trayectorias pierda su realismo en cuanto a la representación visual y háptica del avance del anillo; éstas no deben tener un sentido circular. Si las trayectorias presentaran este comportamiento, no sería posible representar el avance perpendicular del anillo respecto al sentido de la trayectoria; porque

implicaría que el efector final de la interfaz Omni Phantom deba girar libremente sobre el eje z, lo cual no puede hacer debido a que tiene un campo de rotación de 280 grados sobre el mencionado eje. Por lo tanto las trayectorias pueden presentar curvas pero que nunca se cierren sobre sí mismas.

Otra de las razones por las que las trayectorias no deben ser estrechamente cerradas es porque cuando se calcula el punto más cercano sobre éstas con referencia a la posición actual del cursor háptico, éste punto puede cambiar de un lado de la trayectoria a estar inesperadamente al lado opuesto y se presenta una discontinuidad notable en la orientación y magnitud del campo de fuerza, y por lo tanto se dificulta el seguimiento de la misma.

El campo de fuerza utilizado en este ambiente virtual es aplicado en dirección al centro de la trayectoria con el fin de atraer el cursor háptico y de esta manera guiar y apoyar a quien está realizando el entrenamiento. El objetivo es simular un sistema de resorte que aplica una fuerza proporcional a la distancia entre el punto indicado por el cursor háptico y el punto más cercano a éste ubicado sobre la trayectoria.

En el desarrollo de este capítulo se describe el método adoptado para el modelado de las trayectorias utilizadas en el ambiente virtual, y el cálculo del campo de fuerza destinado a apoyar al paciente en la realización de los ejercicios de entrenamiento.

3.1 Elección del método de trazado de trayectorias

La implementación de las trayectorias en el ambiente virtual implica que se haya realizado un modelado matemático de éstas, pues es necesario conocer de manera precisa cada punto para realizar el control de seguimiento con el dispositivo háptico.

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.

Para solucionar este inconveniente existen métodos manejables y compactos para representar una curva. La aproximación consiste en utilizar funciones de mayor grado que las funciones lineales (segmentos rectos). Por ejemplo, si se tienen tres puntos, no se puede obtener una recta que pase por todos ellos a menos de que sean colineales. Esto se debe a que la expresión para una recta

(3.1), sólo dispone de dos coeficientes (a y b) para intentar cumplir tres condiciones; por lo tanto, sería necesario calcular dos rectas.

$$y(x) = ax + b \quad (3.1)$$

Sin embargo, con un polinomio de grado dos (3.2), se dispone de tres coeficientes con los que sí se puede calcular una única función que pase por los tres puntos. Es por eso, que se logra aproximar mejor y con menos información, un segmento de curva mediante una función de mayor grado que con rectas sucesivas.

$$y(x) = ax^2 + bx + c \quad (3.2)$$

Este método es llamado interpolación polinómica, que a partir de un conjunto de puntos de la forma $\{(x_1, y_1) \dots (x_i, y_k) \dots (x_n, y_n)\}$, donde n es la cantidad de puntos, los cuales cumplen la condición de tener abscisas diferentes, es decir, $x_i \neq x_k$ si $i \neq k$; busca un polinomio interpolador de orden $n-1$ o menor, que contenga a la totalidad de los puntos proporcionados para su construcción.

La construcción del polinomio interpolador se basa en que el conjunto de polinomios de grado menor o igual a $n-1$, es un espacio vectorial de dimensión n . De esta manera, dada una base cualquiera de dicho espacio vectorial, el objetivo es hallar las coordenadas del polinomio interpolador respecto de la citada base. El método más sencillo para construir el polinomio interpolador se fundamenta en la base más conocida que es la familia de n polinomios formada por (3.3).

$$\{1, x, x^2, \dots, x^{n-1}\} \quad (3.3)$$

Así, el polinomio interpolador se puede expresar en esta base como se muestra en la fórmula (3.4).

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} \quad (3.4)$$

Por lo que se deben hallar las n incógnitas $\{a_0, a_1, \dots, a_{n-1}\}$.

Como el polinomio pasa por todos los puntos, entonces se debe verificar que al introducir el valor de la abscisa en la variable, se obtenga la ordenada correspondiente (3.5).

$$\begin{aligned}
p(x_1) &= a_0 + a_1x_1 + a_2x_1^2 + \dots + a_{n-1}x_1^{n-1} = y_1 \\
p(x_2) &= a_0 + a_1x_2 + a_2x_2^2 + \dots + a_{n-1}x_2^{n-1} = y_2 \\
&\vdots \\
p(x_i) &= a_0 + a_1x_i + a_2x_i^2 + \dots + a_{n-1}x_i^{n-1} = y_k \\
&\vdots \\
p(x_n) &= a_0 + a_1x_n + a_2x_n^2 + \dots + a_{n-1}x_n^{n-1} = y_n
\end{aligned} \tag{3.5}$$

La fórmula (3.5), es un sistema de n ecuaciones y n incógnitas; y los coeficientes del polinomio interpolador se obtienen resolviendo este sistema [26].

La implementación de un sistema de ecuaciones como (3.5), en la aplicación del Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior, no es recomendable porque es necesario que la fórmula que determina la trayectoria, deba recalcularse cada vez que se determina el punto más cercano sobre ésta, con respecto a la posición del cursor háptico. Siendo el cursor háptico un objeto en movimiento en el ambiente virtual, su posición cambia continuamente y el cálculo del punto más cercano sobre la trayectoria se realiza a una frecuencia muy alta.

Por lo tanto, se requiere que la fórmula que define la trayectoria del ambiente virtual no necesite una cantidad elevada de cálculos algebraicos, ya que esto ralentizaría la respuesta del sistema.

Una mejor opción de modelado de las trayectorias es el método de Curvas de Bézier. El nombre de este tipo de curvas se debe a su inventor Dr. Pierre Bézier, quien fue un ingeniero que trabajó para la compañía automovilística Renault y a principios de los años 60 las desarrolló como una formulación de curvas que le guiaran en el diseño gráfico [27].

Para la creación de una curva de Bézier se necesita 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 curvas de Bézier utiliza una base diferente a la expuesta en (3.3). Esta base la constituyen los polinomios de Bernstein, cuya expresión se muestra en (3.6).

$$B_i^N(s) = \binom{N}{i} (1-s)^{N-i} s^i \quad (3.6)$$

$$\binom{N}{i} = \frac{N!}{i!(N-i)!} \quad (3.7)$$

Donde (3.7) es el coeficiente binomial del Teorema del Binomio de Newton y N es el orden de la curva. Adicionalmente en la fórmula de las curvas de Bézier se incluyen los valores de los vértices de control P_i , obteniéndose como resultado la fórmula (3.8); en la que los valores que toma la variable s , se encuentran en el intervalo cerrado comprendido entre 0 y 1.

$$B(s) = \sum_{i=0}^N B_i^N(s) P_i \quad s \in [0,1] \quad (3.8)$$

A continuación se describe el proceso para la obtención de la fórmula de una curva de Bézier lineal (3.9), partiendo de (3.8), con puntos de control P_0 y P_1 , en \mathbb{R}^3 :

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

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

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

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

$$B(s) = (1-s) \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + s \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad s \in [0,1] \quad (3.9)$$

Para una curva de Bézier lineal, con dos puntos de control, P_0 y P_1 ; la variable s puede interpretarse como un descriptor de cuán lejos se encuentra $B(s)$ en su recorrido de P_0 hacia P_1 . Por ejemplo, cuando s sea igual a 0.25, $B(s)$ es un cuarto de la longitud entre P_0 y P_1 ; este comportamiento se observa en la Figura 3.1.

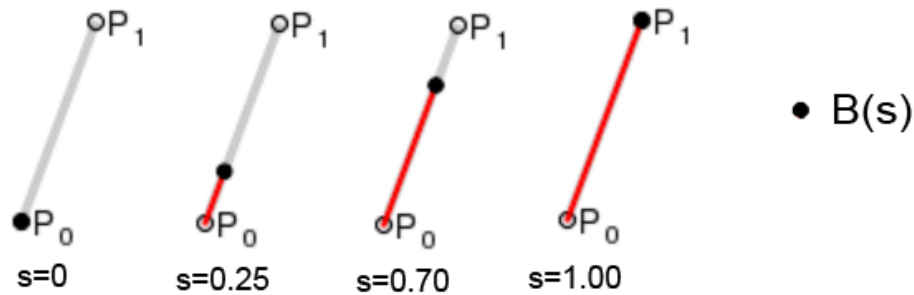


Figura 3.1. Recorrido que realiza $B(s)$ cuando la curva de Bézier es de primer orden⁵.

En una curva de segundo orden, con puntos de control P_0 , P_1 y P_2 ; la construcción de $B(s)$, parte de las dos rectas que conforman el polígono de control, que van desde el punto P_0 hasta P_1 y de P_1 a P_2 . A medida que se incremente la variable s , un punto Q_0 se desplaza sobre la primera recta del polígono de control y un punto Q_1 por la segunda. $B(s)$, se desplaza desde Q_0 hasta Q_1 describiendo una curva cuadrática, como se muestra en la figura 3.2.

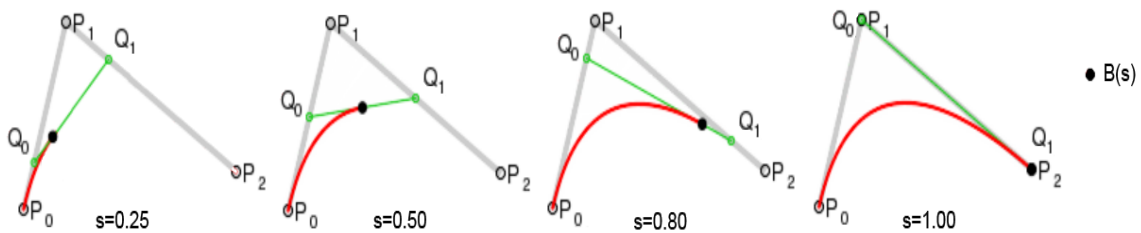


Figura 3.2. Recorrido que realiza $B(s)$ cuando la curva de Bézier es de segundo orden⁶.

Para las curvas de Bézier de orden mayor, se localizan puntos intermedios que van incrementando de orden desde el polígono de control hasta que se alcanza el comportamiento de $B(s)$ en el orden deseado. Por ejemplo, para la construcción de una curva de cuarto orden con vértices de control P_0 , P_1 , P_2 , P_3 , P_4 ; se pueden ubicar los puntos intermedios Q_0 , Q_1 , Q_2 y Q_3 , que describen un movimiento lineal. Sobre estos, se ubican los puntos intermedios R_0 , R_1 y R_2 que describen trayectorias cuadráticas; al mismo tiempo, los puntos S_0 y S_1 se

⁵ Animated linear Bézier curve. Philip Tregoning, 27 August 2006.

⁶ Animated quadratic Bézier curve. Philip Tregoning, 27 August 2006.

ubican entre los puntos R_s presentando el comportamiento de una curva de tercer orden. Finalmente la curva de Bézier de cuarto orden $B(s)$, se mueve entre los puntos S_0 y S_1 . Los puntos de control e intermedios para la construcción de esta curva se observan en la Figura 3.3.

Para implementar las trayectorias en el Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior, es más conveniente la utilización del método de curvas de Bézier por su simplicidad en los cálculos necesarios para la construcción de la misma. Comparado con otros métodos, las curvas de Bézier no necesitan almacenar una gran cantidad de coordenadas, ni resolver un sistema de ecuaciones o que su implementación se realice por tramos de trayectorias pequeñas para construir una de mayor tamaño.

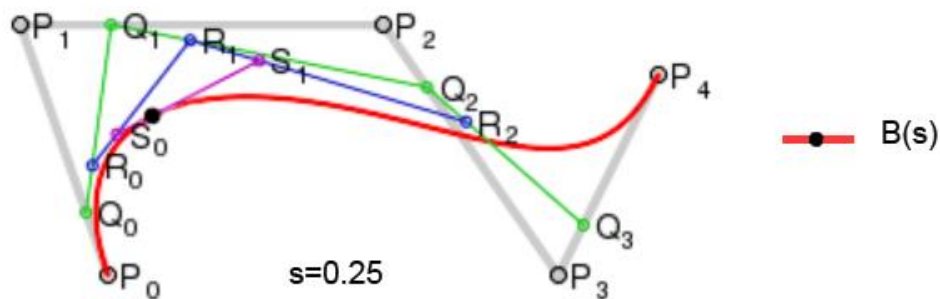


Figura 3.3. Puntos de control e intermedios para la construcción de una curva de Bézier de cuarto orden⁷.

3.2 Curvas de Bézier para las trayectorias de la aplicación

El algoritmo de curvas de Bézier fue implementado en Matlab para fácilmente reproducir distintas curvas de las cuales se eligieron 5 para los niveles de entrenamiento del ambiente virtual. El número de vértices para el polígono de control de las curvas de Bézier se eligió después de realizar pruebas para determinar la cantidad de estos vértices con la que se obtenía un perfil más favorable de la trayectoria.

Si el número de vértices se establecía en un valor comprendido entre 3 y 6, la trayectoria resultante presentaba pocos cambios de dirección (generalmente una única curva suave) y por lo tanto al implementarla en el ambiente virtual no representaría una dificultad aceptable para promover el esfuerzo del paciente al momento del entrenamiento. De 7 hasta 15 vértices de control, las trayectorias resultantes se ajustan a los requerimientos buscados, pues presentan varias curvas suaves que cuando se ingresan al ambiente virtual permiten percibir la

⁷ Animated fourth order Bézier curve. Philip Tregoning, 27 August 2006.

continuidad del campo de fuerza sobre ellas. A partir de 16 vértices de control en adelante, las trayectorias comienzan a presentar las curvas cerradas no deseadas; porque pueden provocar la discontinuidad en el campo de fuerza descrita anteriormente.

Otros inconvenientes asociados a 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 anteriormente mencionado; se decidió que las curvas de Bézier fueran de orden 10, o lo mismo, que contengan 11 vértices de control. Estos 11 puntos se generan aleatoriamente en el plano xy , dentro de un área delimitada por un radio desde el origen hasta máximo 100 unidades; y tienen una tendencia semicircular para que las trayectorias resultantes no se cierren sobre sí mismas.

Las curvas de Bézier incluyen dos de los vértices dados para generarlas, el vértice inicial y el final. El vértice inicial o $V1$, en este caso, se ubica siempre en el primer cuadrante del plano cartesiano, siendo su coordenada en y igual a cero, mientras que en x puede tomar cualquier valor positivo que se encuentre dentro del rango definido.

A partir del primer vértice, los puntos subsecuentes se van generando en sentido opuesto a las manecillas del reloj hasta finalizar en un punto $V11$ ubicado en el segundo cuadrante del plano cartesiano, con coordenada en y generalmente cercana a cero y en el eje x toma un número negativo mayor que -100 como se observa en la Figura 3.4.

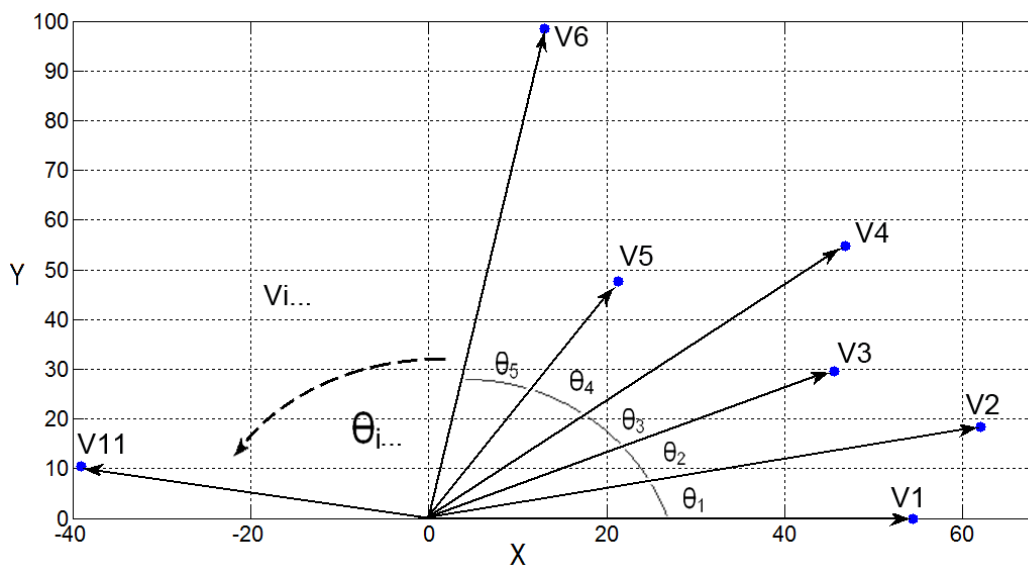


Figura 3.4. Distribución de los vértices de control en el plano xy .

Las magnitudes $|V_i|$ son generadas como números aleatorios con distribución uniforme en un rango de 0 a 100 unidades; y la orientación θ_i , se asigna de manera equidistante entre 0 radianes y π radianes, es decir que los vectores estarán separados 0.288 radianes o 16.5° uno del otro porque hay 11 de ellos. El ángulo θ_i inicia con un valor de 0° y se incrementa cada vez en 16.5° .

Las componentes x e y de los puntos se obtienen con la magnitud y el ángulo de cada vector; siendo la coordenada en x el producto de la magnitud del vector por el coseno del ángulo y la coordenada en y el producto de la magnitud por el seno del ángulo, como se indica en (3.9) y (3.10). De esta forma se obtienen los vértices que conformarán el polígono de control de cada curva de Bézier.

$$x_i = |V_i| * \cos \theta_i \quad (3.9)$$

$$y_i = |V_i| * \sen \theta_i \quad (3.10)$$

La curva del Nivel 1 de entrenamiento ha sido tomada como ejemplo para mostrar su polígono de control generado para su construcción, el cual se muestra en la Figura 3.5.

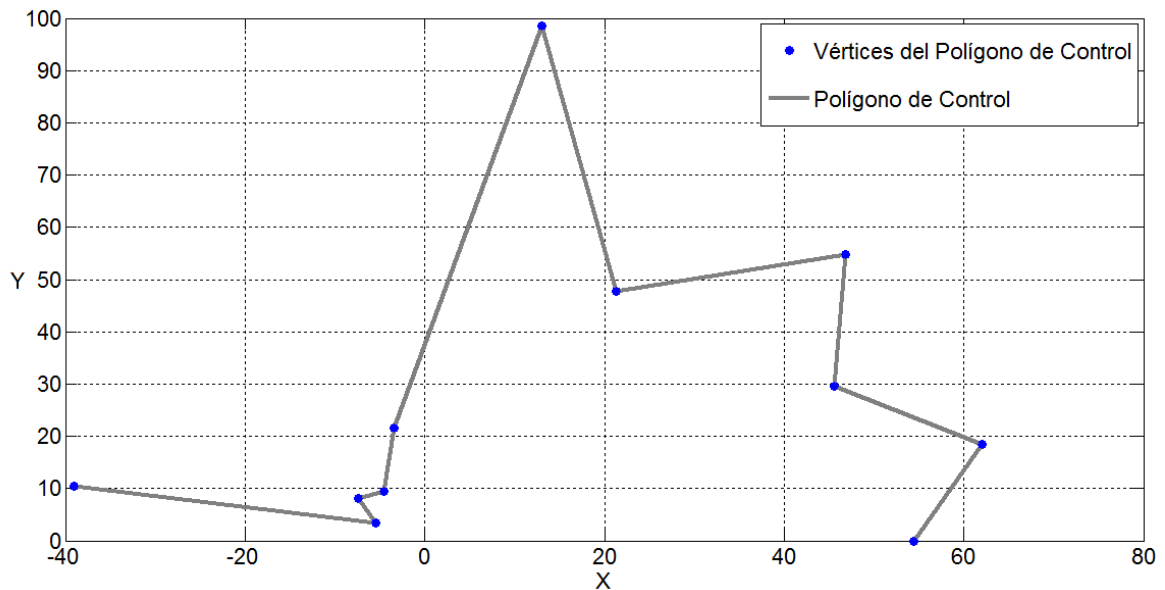


Figura 3.5. Polígono de control de la curva de Bézier del Nivel 1 de entrenamiento.

Los vértices de control son ingresados en la fórmula de curvas de Bézier (3.2) y se calcula la curva resultante, mediante la asignación de valores a la variable s de la fórmula. Como en esta etapa sólo se necesita la apariencia de la curva de Bézier y los once puntos de control, el total de valores que se evaluaron en la fórmula fueron cien puntos. En la Figura 3.6 se observan los vértices de control,

el polígono de control y la curva de Bézier resultante para el Nivel 1 de entrenamiento.

3.3 Apariencia 3D de las trayectorias

En la aplicación de Ambiente Virtual para Rehabilitación de Motricidad Fina de Miembro Superior se cuenta con 5 niveles de entrenamiento que se describen en el Capítulo 4. Cada uno de estos niveles cuenta con una trayectoria de Bézier diferente, que tiene la apariencia de una tubería suspendida en el espacio. Este efecto es logrado mediante el motor de renderizado gráfico OpenGL que se utiliza.

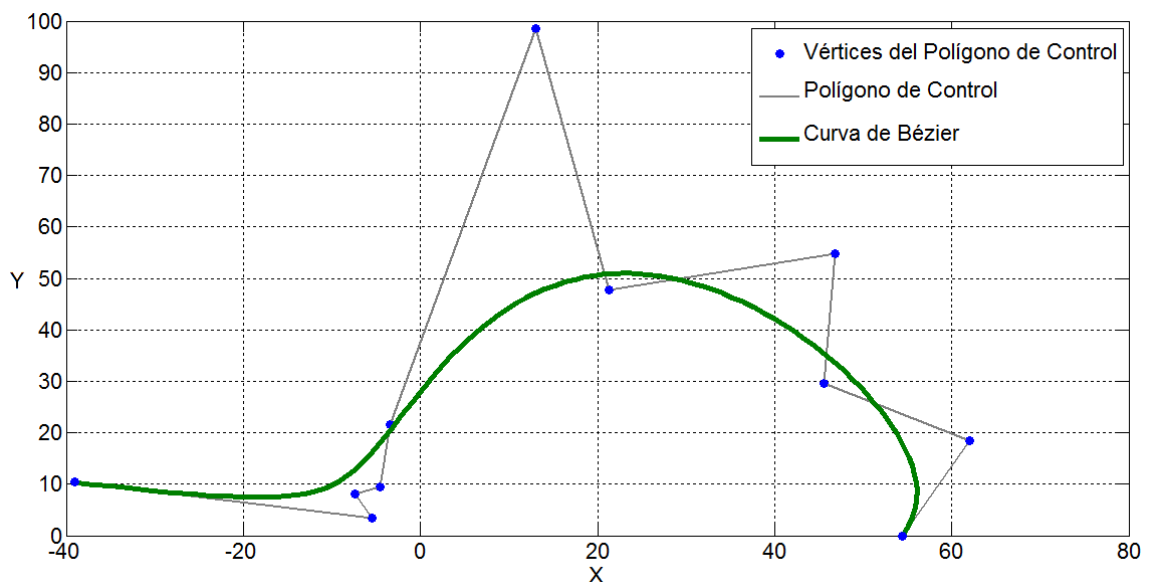


Figura 3.6. Curva de Bézier con su respectivo polígono de control.

OpenGL permite cargar en un ambiente virtual modelos realizados en un programa de Diseño Asistido por Computadora (CAD), siempre que estos modelos se hayan transformado a un formato que OpenGL pueda soportar. El proceso de construcción de las trayectorias, partiendo de la gráfica generada en Matlab (Figura 3.6), hasta un modelo tridimensional que es cargado en el entorno de entrenamiento virtual, se describe en el Anexo B del presente trabajo. Si no se realizara el procedimiento indicado en el Anexo B, la apariencia de las trayectorias en el ambiente virtual que se pudiera lograr sería de una línea y no de una tubería.

Un aspecto clave en OpenGL para que los objetos se miren como sólidos en un ambiente tridimensional es la habilitación y configuración de la iluminación del espacio de trabajo; si esto no se realiza, los objetos se miran solamente de un

color, sin matices que indiquen que la luz sobre ellos llega con un ángulo diferente en cada parte provocando algunas sombras que le dan la apariencia de ser un cuerpo en tres dimensiones. En la sección de la ventana donde los pacientes realizan el entrenamiento, la trayectoria y el anillo para seguirla tienen una apariencia como la que se muestra en la Figura 3.7.

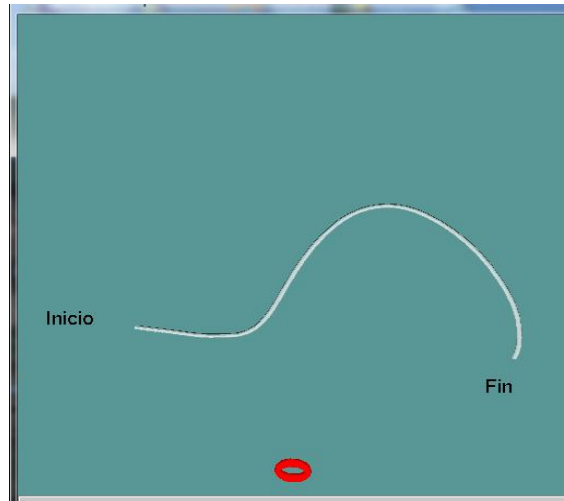


Figura 3.7. Ambiente Virtual de entrenamiento.

3.4 Implementación del modelado matemático de las trayectorias en el ambiente virtual

Para poder lograr el objetivo de la configuración de un campo de fuerza que genere una atracción del efector final de la interfaz háptica Omni Phantom hacia la trayectoria deseada, se debe cargar el modelo matemático de la misma. Es con este modelo que se realizan los cálculos necesarios dependientes de la posición del efector final y de los puntos dentro de la trayectoria, para establecer la dirección y la magnitud de la fuerza.

Con el programa creado en Matlab se eligieron las 5 trayectorias a implementar en los niveles de entrenamiento de la aplicación. Para reproducir estas curvas de Bézier en el ambiente virtual fue necesaria la instauración de un algoritmo en C++ con un fin similar al implementado en Matlab. La diferencia respecto a este último, fue que no necesitaba la generación de puntos como vértices de control porque ya estaban establecidos.

El volumen de visualización de los gráficos de OpenGL y de la escena háptica, se estableció como un cubo con una arista de 400 unidades, con el origen de coordenadas situado en su centro, de modo que los ejes x, y y z se extienden desde -200 a 200 cada uno. Los puntos tomados de Matlab se encuentran en una escala menor al del ambiente virtual; por lo tanto, es necesario realizar un

escalamiento a un tamaño apropiado mediante la multiplicación por una constante, sin que se altere la forma de la trayectoria.

La razón por la que no se efectuó este escalamiento desde Matlab, fijando la magnitud máxima del vector aleatorio como 200 unidades; fue la no obtención de una curva que se ajustara plenamente con el ambiente virtual. Las curvas resultantes quedaban siempre de tamaño menor al requerido. Por otro lado, si se fijaba la magnitud máxima del vector aleatorio a un valor mayor a 200 unidades; las trayectorias resultantes podían quedar, en parte, por fuera del campo de visión en el ambiente virtual. Por las razones anteriores se tomó la decisión de crear la curva de Bézier en pequeña escala y después ajustarla al ambiente virtual.

Los datos escalados adecuadamente se cargan al algoritmo en C++ que calcula la curva de Bézier y permite dar valores a la variable entre 0 y 1, para calcular los puntos que forman la trayectoria. El total de valores entre 0 y 1 que fueron evaluados en la curva de Bézier fue de 300; esta cantidad de puntos es suficiente para crear la sensación de un campo de fuerza continuo a través de ella.

Las pruebas realizadas con una cantidad menor de puntos, presentaron comportamientos no deseados, entre ellos vibración del efector final y sensación de un campo de fuerza incoherente con el seguimiento de la trayectoria en el ambiente virtual.

Al establecer una cantidad mayor de puntos sobre la trayectoria, por encima de 300, la mejora en la sensación de continuidad no es perceptible; así que para reducir el consumo de recursos computacionales durante el funcionamiento del ambiente virtual no se aumentó el número de puntos.

El algoritmo de curvas de Bézier en C++, calcula la trayectoria pero no permite visualizarla en el ambiente virtual, por eso se recurre a las funciones de OpenGL de dibujo de primitivas para poder hacerlo. Esta forma de visualización de la trayectoria es sólo temporal y sirve como ayuda para realizar el ajuste del objeto 3D diseñado en un programa CAD con la trayectoria calculada matemáticamente.

De las múltiples primitivas de dibujo soportadas por OpenGL, tres fueron las opciones para realizar el dibujo de la trayectoria: los puntos, con el comando `GL_POINTS` que ubica en la pantalla el punto dependiendo de sus coordenadas en el espacio definidas en *glVertex3f(x,y,z)* a cuyo argumento se pasan las coordenadas del punto generadas por la curva de Bézier; las líneas segmentadas con `GL_LINES` que igualmente utiliza *glVertex3f* para dibujar, pero toma pares de estas coordenadas que indican el inicio y final de una línea dejando un espacio sin dibujar comprendido entre el vértice final de una línea y el vértice inicial de la siguiente; y por último las líneas continuas con

GL_LINE_STRIP que traza una línea desde el primero al último de los vértices creando una mejor apariencia de la trayectoria, razón por la que fue elegida.

Cuando la visualización de la curva de Bézier en el ambiente virtual ya se encuentra establecida, se carga el modelo 3D correspondiente, que en un principio no coincidirá con la trayectoria dibujada pero que se debe ajustar variando su escala en x y en y.

Como el modelo 3D de la curva de Bézier fue creado de un perfil que representa fielmente la curva original, se tiene certeza de que calce sobre la curva dibujada en el ambiente virtual. Después de los ajustes de escalamiento y posición, se deshabilita la trayectoria dibujada con la primitiva GL_LINE_STRIP de OpenGL, pues se estaría dibujando inútilmente ya que si el ajuste es preciso, el objeto 3D la oculta totalmente.

3.5 Campo de fuerza de asistencia al seguimiento de las trayectorias

Las trayectorias en el ambiente virtual se representan como tuberías en el plano xy , y son recorridas por el cursor de la interfaz háptica que se representa por un toroide con un diámetro que permite ver en la escena gráfica que se puede introducir la trayectoria en él. El campo de fuerza, se encarga de la atracción del proxy, o punto de contacto ubicado en el centro del toroide que se ha cargado como cursor, hacia el centro de la trayectoria; para poder lograr esto se necesitan realizar varias etapas que se describen a continuación:

3.5.1 Configuración del Cursor Háptico

La finalidad del cursor háptico es la representación de la posición y orientación del efector final de la interfaz háptica en el ambiente virtual y permite observar que la sensación táctil que provee la interfaz se corresponda con una escena en el ambiente gráfico. Por ejemplo, cuando ocurre la realimentación de fuerza por parte del dispositivo y crea la sensación de una barrera infranqueable el cursor háptico debe representar en el ambiente virtual que efectivamente está colisionando con la cara de un objeto sólido con el área suficiente como para detener el avance del cursor.

Como las librerías de programación del dispositivo están diseñadas para trabajar de manera predeterminada con el motor de renderizado gráfico OpenGL, las formas que generalmente se asocian para representar el cursor son los objetos 3D definidos en la librería GLUT, basada en OpenGL; dichos objetos son: una esfera, un toroide, un cubo, un cono, un cilindro y una tetera.

Sin embargo, mediante el mismo método descrito en el Anexo B, para cargar las trayectorias con apariencia 3D en el ambiente virtual, es posible construir un objeto sólido personalizado y asociarlo al cursor háptico.

El toroide cargado como cursor háptico en la aplicación tiene la capacidad de girar y trasladarse en los tres ejes del ambiente virtual de forma consecuente con la posición y giro del efector final de la interfaz háptica. Para lograr esto se tiene una función llamada *DibujarCursor* encargada de obtener las coordenadas y orientación actuales del efector final del dispositivo y asociarlas al objeto 3D cargado en el ambiente virtual para representarlo. Si el objeto 3D cargado fuera una esfera, no sería necesaria la obtención de la orientación del efector final porque al asociarla a la esfera no sería perceptible los giros, sólo importaría la posición que sería obtenida mediante el uso de la siguiente instrucción de la librería HD:

```
hduVector3Dd posicion;  
hdGetDoublev(HD_CURRENT_POSITION, posicion)
```

Que obtiene la posición del efector final del dispositivo y la retorna como un vector en R^3 cuyos valores en las posiciones 0,1 y 2 se corresponden a la coordenada en los ejes x , y y z respectivamente. Si se asocia este vector al objeto 3D, y si es un toroide, puede observarse que se desplaza de manera consecuente con el efector final del dispositivo; pero si éste gira, los giros no pueden verse en el toroide que se mueve en el ambiente con una misma orientación. Por lo tanto esta instrucción no es la más adecuada para la configuración del cursor háptico, pero sí lo será más adelante en el cálculo de la fuerza aplicada por el dispositivo.

Para poder obtener, y después asociar la orientación del efector final del dispositivo es necesaria otra instrucción, que obtiene la matriz de transformación del cursor:

```
HLdouble proxyxform[16];  
hlGetDoublev(HL_PROXY_TRANSFORM, proxyxform);
```

En esta matriz está contenida la posición y la orientación del efector final de la interfaz háptica. Esta matriz es utilizada por una función de OpenGL que la asocia al objeto 3D, mediante la multiplicación por la matriz que representa el estado del cursor.

3.5.2 Determinación del punto más cercano de la curva respecto a la posición del cursor háptico.

El campo de fuerza del ambiente virtual ejerce su función de aplicar una fuerza proporcional a la distancia entre el cursor háptico y la trayectoria, en un único punto de la trayectoria a la vez. El punto sobre la trayectoria hacia el que se dirige la fuerza, es el que se encuentra a una distancia menor respecto al cursor háptico, en un instante determinado. Para establecer cuál de los 300 puntos de la trayectoria cumple con esa condición, es necesaria la implementación de un algoritmo para que realice el cálculo.

Se probaron dos algoritmos de minimización para hallar el punto sobre la trayectoria que minimiza la distancia hacia la posición del cursor háptico. Estos algoritmos corresponden al método de Brent y a un método exhaustivo.

Método de Brent

El método de Brent es un algoritmo de minimización que combina la interpolación cuadrática inversa y los métodos de bisección y de la secante; estos dos últimos para generar el siguiente intervalo de interpolación. Para la búsqueda del punto más cercano a la posición del proxy mediante este método se debió minimizar la función conformada por las distancias calculadas hacia los puntos de la trayectoria.

Mediante el método de bisección se proporcionaron los intervalos en los que se realiza la búsqueda del mínimo. Como la curva de Bézier se construye con los valores de la variable s , entre 0 y 1, y el cursor háptico podría encontrarse cerca de cualquiera de sus puntos, el primer intervalo proporcionado es $[0,1]$, que comprende toda la curva de Bézier. Un tercer valor es fijado en 0.5 para iniciar las iteraciones. Este valor es producto de la fórmula (3.11), del método de bisección, donde a_0 y b_0 son los límites del intervalo.

$$c_0 = \frac{a_0 + b_0}{2} \quad (3.11)$$

Los tres valores se evalúan en la curva de Bézier y se calcula sus distancias hacia la posición del cursor háptico, con la fórmula (3.12). Donde $B(s)$, es la fórmula de la curva de Bézier (3.8) y p_x , p_y y p_z , la posición del cursor háptico.

$$Distancia = \left\| B(s) - \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \right\|^2 \quad (3.12)$$

Con las tres parejas de valores conformadas entre 0, 0.5, 1 y sus respectivas distancias, se calcula una parábola a la que se le halla el punto mínimo. La abscisa de este punto mínimo corresponde a un valor ubicado entre 0 y 1, el cual se evalúa en la curva de Bézier y se calcula la distancia hacia la posición del cursor háptico, esta distancia irá reduciendo a medida que suceden las iteraciones. Con el mismo valor de la abscisa del punto mínimo se calcula un nuevo intervalo de búsqueda.

Estas operaciones se repiten hasta que se alcanza una tolerancia que indica la parada del algoritmo. Cuando esto sucede, los valores que se retornan son la última distancia calculada y su respectivo punto sobre la trayectoria de Bézier; los cuales son utilizados para calcular la magnitud y dirección de la fuerza.

Método exhaustivo

Este método consiste en realizar el cálculo de las distancias (3.12), para todos los puntos de la curva de Bézier respecto a la posición del cursor háptico. A medida que se calculan las distancias, se realiza una comparación de la distancia calculada con la distancia anterior; la de menor valor se almacena hasta que otra distancia mucho menor (si la hay), la desplaza tomando su lugar. Al tiempo que la distancia es almacenada, se hace lo mismo con el valor de la variable s que produjo esa distancia. Al final del cálculo de todas las distancias, se obtendrá la que fue de menor valor junto con el punto sobre la curva de Bézier más cercano a la posición del cursor háptico. Estos dos valores son enviados a la función que calcula la magnitud y dirección de la fuerza.

Después de probar estos dos métodos en el ambiente virtual, se pudo comparar el rendimiento de ambos, presentándose como resultado más favorable la utilización del método exhaustivo; por lo cual, fue el que finalmente se adoptó.

3.5.3 Cálculo del Campo de Fuerza

El campo de fuerza implementado simula un sistema de resorte que atrae el cursor háptico hacia la trayectoria con una fuerza proporcional a la distancia; la fórmula (3.13) representa este sistema, en la que K , es la constante de proporcionalidad del resorte simulado y d , es la distancia cuyo cálculo se describió anteriormente.

$$\overrightarrow{Fuerza} = -K * d \quad (3.13)$$

En la Figura 3.8, se muestra la trayectoria y el anillo, cuyo centro CA, no se encuentra alineado respecto al eje de la trayectoria CT, por lo tanto se aplica

una fuerza FA en dirección a CT para corregir la desviación. Esta fuerza es similar a la que se sentiría al tener el centro del anillo sujeto al centro de la trayectoria mediante un resorte, mientras mayor sea la desviación, mayor será la fuerza ejercida por este.

La constante de proporcionalidad K de la fórmula (3.13), varía dentro de un rango que se define dependiendo de las características de generación de fuerza de la interfaz háptica Omni Phantom. El rango que se presenta en pantalla para la variación de asistencia va desde 0 hasta 1, esta variable se asocia internamente a la constante de proporcionalidad de la fórmula del campo de fuerza. El campo de fuerza deja de aplicarse cuando el grado de asistencia es 0, y aplica la fuerza máxima calculada para que el dispositivo tenga un funcionamiento seguro cuando el grado de asistencia es 1.

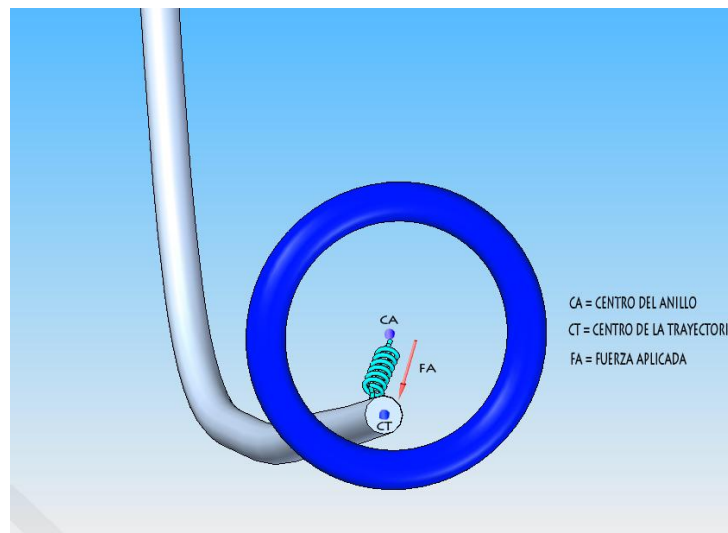


Figura 3.8. Representación del sistema de resorte encargado de aplicar la fuerza.

Otro componente de la fórmula (3.13), es la distancia d , que se introduce al cálculo del campo de fuerza y que después de que se configura el grado de asistencia, es ésta la que determina la magnitud de la fuerza ejercida por el dispositivo háptico.

El cálculo de la dirección en la que se debe aplicar la fuerza, es una de las partes más importantes de la aplicación, ya que es ésta la que permite guiar el cursor háptico hacia el punto adecuado dentro de la trayectoria. Como el cursor háptico se mueve a través del espacio, las direcciones que pueda tomar la fuerza que se ejerce sobre él, comprenden cualquiera ubicada en el espacio tridimensional. En la Figura 3.9 se presenta un ejemplo de las direcciones que puede tener la fuerza calculada sobre la trayectoria.

El algoritmo que calcula la menor distancia entre el cursor háptico y un punto sobre la trayectoria, retorna además de la magnitud de la distancia, las coordenadas del punto de la trayectoria que cumplió con esa condición, de modo que se tiene un vector ubicado en el espacio. Por otro lado, el uso de una instrucción de la librería HD permite conocer la posición del cursor háptico y la retorna en un vector en R^3 con las coordenadas x, y y z de esta posición; de esta forma se tienen dos vectores en el espacio que se utilizarán para obtener la dirección de la fuerza.

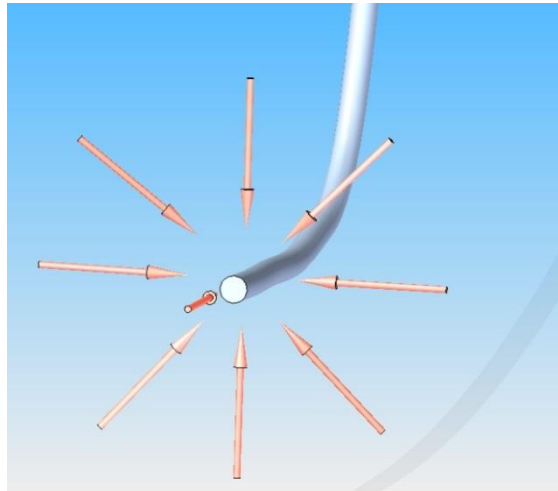


Figura 3.9. Posibles direcciones de la fuerza ejercida por el dispositivo háptico.

Se realiza la diferencia entre estos dos vectores; al Vector 1 (punto sobre la trayectoria) se le resta el Vector 2 (posición del cursor háptico) y el resultado será un Vector 3 dirigido desde el Vector 2 hacia el Vector 1, que es la dirección que se necesita. La forma como se pueden ubicar estos tres vectores se observa en la Figura 3.10.

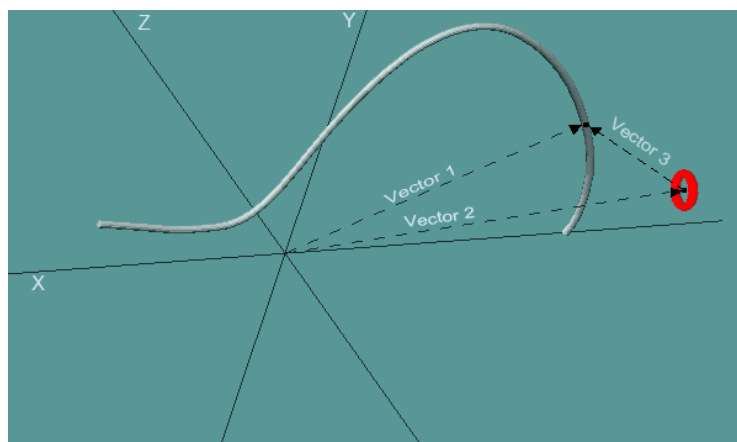


Figura 3.10. Vector 1: punto sobre la trayectoria. Vector 2: posición del cursor háptico. Vector 3: diferencia.

El Vector 3 tiene la orientación requerida para aplicar la fuerza del dispositivo háptico, pero cuenta con su propia magnitud, por lo tanto no es posible multiplicarlo directamente por la magnitud de la fuerza porque la distorsionaría. Lo que se debe hacer es normalizarlo para que tenga un valor de 1 y al momento de multiplicarlo por la magnitud de la fuerza, sólo le aporte la dirección. Esta normalización se hace mediante la utilidad de la librería HD para vectores llamada *normalize*, y su modo de uso se muestra a continuación donde *unitdiferencia* es el vector unitario buscado:

```
hduVector3Dd diferencia;  
diferencia = cercanoPoint - pos;  
hduVector3Dd unitdiferencia = normalize(diferencia);
```

En este punto ya se tiene implementado el campo de fuerza de asistencia al seguimiento de la trayectoria, pero si se corre el programa del ambiente virtual, se advierte que al anillo no es guiado exactamente a la trayectoria, sino que su centro queda un poco por debajo del centro de ella; aún con el mayor nivel de fuerza aplicado puede verse la variación.

Esta variación en el guiado del campo de fuerza se debe al efecto de la gravedad sobre el brazo del dispositivo háptico. Si esto no se corrigiera, el ambiente virtual no presentaría un comportamiento consecuente con los movimientos que el usuario realiza en el mundo real con el dispositivo háptico y se perdería la interactividad.

Otro inconveniente asociado a la no compensación de la gravedad, consiste en que el usuario presente fatiga muscular, debida a que tiene que sostener con su propia fuerza el peso del brazo del dispositivo cuando aún no se ha activado el campo de fuerza.

Se necesitó crear un vector de magnitud constante, que actuara sobre el eje y del ambiente virtual en sentido contrario a la gravedad. Para poder ajustar la magnitud, se desactivó el campo de fuerza y en su lugar se retornaba el vector que estaba siendo ajustado. La calibración terminó cuando al posicionar el efector final en cualquier lugar del espacio de operación de la interfaz háptica, éste permanecía inmóvil en ese lugar.

La fórmula definitiva del campo de fuerza (3.14), fue la que además de calcular la fuerza proporcional a la distancia y tener asociada una variable para cambiar el valor de la constante de proporcionalidad, contenía la adición del vector antigravedad.

$$\overrightarrow{Fuerza} = 0.5 * GradoAsistencia * unitdiferencia * Distancia + Antigravedad \quad (3.14)$$

El factor 0.5 en la fórmula (3.14), se encarga de limitar la magnitud de la fuerza para que el funcionamiento del dispositivo háptico sea seguro y no presente ningún riesgo para sus componentes internos. Este factor limita la fuerza del dispositivo cuando el cursor háptico se encuentra en una región muy cercana alrededor de la trayectoria, donde si se aplicara una fuerza mayor, se presentarían vibraciones en el dispositivo háptico.

La implementación final dio como resultado un campo de fuerza continuo alrededor de la trayectoria a seguir mediante el cursor háptico, que cumple la condición de sinteticidad de los ambientes virtuales (Capítulo 1); y que permite un funcionamiento seguro tanto para los componentes hardware del dispositivo, como para el usuario que utiliza la aplicación.

Capítulo 4

Descripción del prototipo desarrollado

En este capítulo se describen las características de funcionamiento del prototipo desarrollado en este trabajo; el cual se divide en una interfaz de usuario y un ambiente virtual. La interfaz de usuario consta de tres formularios principales; en el primero se ingresan los datos personales del paciente, se hace la evaluación del mismo y finalmente se cuenta con un campo destinado para que el profesional encargado realice las observaciones pertinentes acerca del estado del paciente. El segundo formulario permite que se realicen consultas sobre el paciente referentes a sus datos personales, las evaluaciones que se le han realizado y observar el historial de desempeño en la realización de los ejercicios en el ambiente virtual. En el tercer formulario se listan todos los datos personales de los pacientes registrados; adicionalmente se puede consultar los datos de un paciente en particular, buscado a través de su número de cédula o su nombre. El ambiente virtual contiene dos subdivisiones, la primera son ejercicios de familiarización con el entorno háptico y la segunda son los cinco ejercicios de entrenamiento, cada uno con una trayectoria diferente y con la implementación del campo de fuerza de asistencia al seguimiento de estas trayectorias. En el Anexo C, se muestra el procedimiento de instalación de la aplicación con todos sus requerimientos.

4.1 Descripción de la Interfaz de Usuario

Existen varias maneras de acceder a los tres formularios con los que cuenta la aplicación (ver Figura 4.1). En la primera se selecciona la opción 'Archivo' desde la barra de menú de la parte superior de la pantalla, al hacerlo se desplegarán las opciones posibles, finalizando con la opción 'Salir'. Otra manera de abrir estos formularios es haciendo clic en la pestaña correspondiente, que podría ser: 'Registro Paciente', 'Consultar Registro' y

‘Ver Todos los Registros’, la cual está ubicada en la parte superior de la pantalla, debajo de los menús. Finalmente, está la opción de escoger cualquiera de los formularios dando clic en alguno de los iconos ubicados arriba de las pestañas.

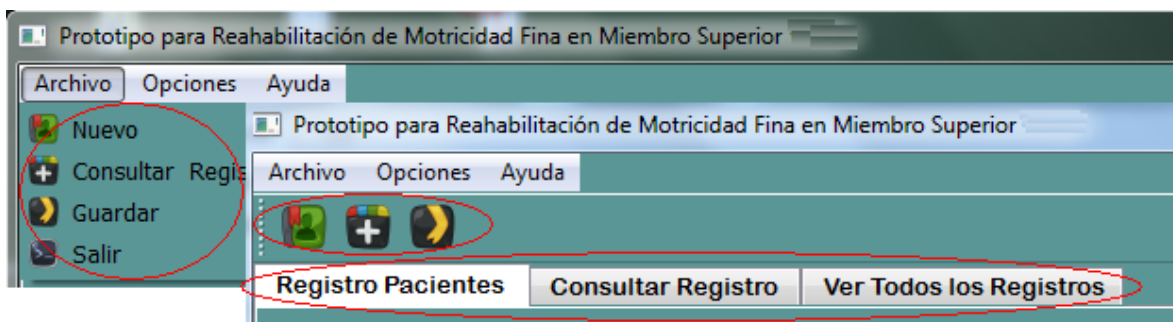


Figura 4.1. Maneras de acceder a formularios de la aplicación.

4.1.1 Formulario uno, ‘Registro de Pacientes’

Normalmente las instituciones de salud manejan formatos en los que registran la información personal de los pacientes y también diligencian listas de chequeos y/o evaluaciones para medir el desempeño de las habilidades motrices, la información aquí solicitada es propia de cada institución. En el caso de la aplicación desarrollada, esta información se encuentra en el formulario uno, ‘Registro de Pacientes’ (ver Figura 4.3). Los datos personales, ubicados en la parte superior (número uno) corresponden a los comúnmente usados y son: número de documento, tipo de documento, nombre, procedencia, ocupación, sexo, edad, teléfono, fecha de evaluación y evaluador. En el momento en el que el fisioterapeuta o persona encargada del manejo de la aplicación esté introduciendo los datos en mención, los únicos campos obligatorios que se deberán diligenciar son los que corresponden al número de documento y al nombre del paciente, de no hacerse no se guardará el registro y aparecerá un mensaje realizando la advertencia (ver figura 4.2).

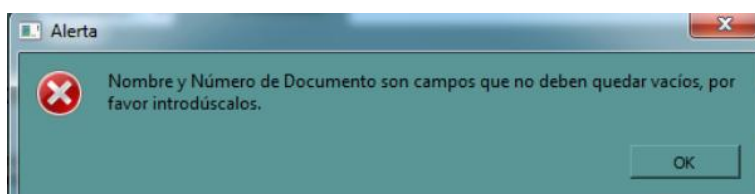


Figura 4.2. Mensaje de alerta generada por campos vacíos.

En la parte dos del formulario está la evaluación del paciente, la cual fue tomada de una propuesta sobre evaluación curricular de la motricidad y la autoayuda, adelantada en la Universidad Autónoma de Madrid, dentro de la Facultad de Psicología [10]. En ella, entre otras cosas se propone una metodología para evaluar las habilidades referidas a la motricidad fina, dicha evaluación está compuesta por 15 ítems, los cuales pueden observarse en la Tabla 4.1. El propósito es que el paciente realice una serie de actividades que son propias de la motricidad fina; el evaluador debe seleccionar las que considere convenientes, de acuerdo a la edad del paciente y características del mismo que se hayan determinado con anterioridad. Del desempeño en la ejecución de las mismas depende el diagnóstico que haga el fisioterapeuta respecto al estado del paciente.

Archivo Opciones Ayuda

Registro Pacientes Consultar Registro Ver Todos los Registros

IDENTIFICACIÓN

Número Documento Tipo de Documento

Nombre

Procedencia Ocupación

Sexo Edad Teléfono

Fecha de evaluación 25/10/2012 Evaluador

EVALUACIÓN DEL PACIENTE

Seleccione una a una las categorías y responda a las preguntas

Agarrar y sostener
Alcanzar
Anudar
Apilar
Arrojar
Atornillar y desatornillar
Conductas reflejas y comportamier

Amontona bloques de cualquier forma
Construye torres de dos cubos o cinco
Construye torres de más de 6 cubos

Observación

Agregar Registro

Figura 4.3. Interfaz de Usuario, Formulario uno, 'Registro Pacientes'.

| DIMENSIÓN DE LA MOTRICIDAD FINA | ACTIVIDADES |
|--|---|
| Conductas reflejas y comportamientos básicos | Cierra las manos formando un Puño. |
| | Es capaz de extender los dedos, tocando con el pulgar cada dedo cuando se le ayude físicamente. |
| Agarrar y sostener | Sostiene objetos pequeños que se le ponen dentro de o sobre la palma de la mano durante 5 segundos, por lo menos. |
| | Pasa objetos de una mano a otra. |
| Alcanzar | Alcanza objeto atractivo, que esta fuera de su alcance. |
| Arrojar | Lanza o arroja de cualquier manera objetos pequeños a una distancia de al menos 60 cm. |
| Apilar | Amontona bloques de cualquier forma. |
| | Construye torres de dos cubos o cinco. |
| | Construye torres de más de 6 cubos. |
| Ensartar | Coccola 3 clavijas grandes en el tablero, al azar, coloca clavijas en agujeros consecutivos horizontal y verticalmente. |
| | Inserta el hilo por una aguja de coser. |
| Tirar | Da un tirón o tira de una cuerda. |
| Manejo de Papel | Desenvuelve un caramelo o cualquier otra cosa sin ayuda. |
| | Dobla una hoja de papel por la mitad de forma que los bordes no queden separados más de 2 centímetros. |
| Cortar con tijeras | Utiliza las tijeras para cortar papel de cualquier forma. |
| | Corta en línea recta y curva impresa o en relieve. |
| Anudar | Pone lazos a objetos. |
| | Quita lazos. |
| Empujar | Baja una palanca al empujar con la mano. |
| Girar | Gira un pomo (por ejemplo el de la puerta) con una mano. |
| | Da cuerda a un juguete. |
| Verter | Vierte un cubo que está en una taza a otra taza. |
| | Vierte agua o arena de una taza a otra. |
| Atornillar y desatornillar | Abre tapones de tapa de rosca. |
| | Cierra tapones de tapa de rosca. |
| Movimientos manuales de precisión | Precisión en lanzar objetos con gran puntería. |
| | Precisión en clavar con un martillo. |
| | Precisión al atarse el nudo de la corbata. |

Tabla 4.1. Dimensiones de la Motricidad Fina [10].

En la parte dos del formulario, el fisioterapeuta seleccionará los ítems, en el lado izquierdo de la pantalla, en el derecho se mostrarán las actividades correspondientes, que adicionalmente cuentan con una lista desplegable, al dar clic sobre ésta, se mostrarán las opciones posibles, el evaluador elegirá la que considere adecuada (Ver figura 4.4).

EVALUACIÓN DEL PACIENTE

Seleccione una a una las categorías y responda a las preguntas

Agarrar y sostener
 Alcanzar
 Anudar
 Apilar
 Arrojar
 Atornillar y desatornillar
 Conductas reflejas y comportamientos

• Sostiene objetos pequeños que se le ponen dentro de o sobre la palma de la mano durante 5 segundos, por lo menos.

 Si
 No

S de una mano a otra.

Figura 4.4. Selección de ítem y respuesta a actividades.

De acuerdo al cuadro patológico de un paciente, es posible que el fisioterapeuta considere que los ítems no son suficientes y/o adecuados para evaluar al paciente, en ese caso en la parte final del formulario hay un campo destinado para que se introduzca comentarios respecto a pruebas adicionales que haya realizado el paciente, o cualquier tema que crea conveniente.

El objetivo general de la interfaz de usuario es que se pueda registrar toda la información pertinente respecto a un paciente, esto con la intención de que se pueda tener trazabilidad del desempeño y/o evolución del mismo, además de que se sistematiza el manejo de los datos; aunque se considere lo anterior, de ninguna manera se pretende que la gestión de la información sea obligatoria, el usuario podría ir directamente a realizar los ejercicios de entrenamiento si así se considera adecuado, sin embargo, en el caso de que desee guardar la información del desempeño en la realización de ejercicios, sí es necesario que haya un previo registro de la información en este formulario.

Todos los campos correspondientes a datos personales, evaluación y observación que no sean diligenciados, quedaran registrados en blanco, lo cual se evidenciara en el formulario dos, 'Consultar Registro', éste puede observarse en la Figura 4.5.

Es posible que el fisioterapeuta a medida que avance el proceso de rehabilitación considere que es necesario hacer nuevas evaluaciones para determinar la evolución y/o estado del paciente. Una vez se hayan registrado los datos personales del paciente estos pueden asociarse a varias evaluaciones y observaciones, para ello se puede ingresar el número de documento en el campo correspondiente y a continuación presionar la tecla Enter, al hacerlo si ya se ha registrado previamente el número de documento los datos del paciente asociados al mismo se podrán visualizar en pantalla; se ha dispuesto así para que el usuario pueda agilizar la gestión de los registros.

La aplicación no permitirá guardar datos repetidos del campo de número de documento, los datos personales de algunos pacientes se podrían repetir, lo que no permitirá la aplicación es que haya números de documentos redundantes. Al intentar guardar un número de documento repetido se mostrará un mensaje, este puede verse en la figura 4.6.

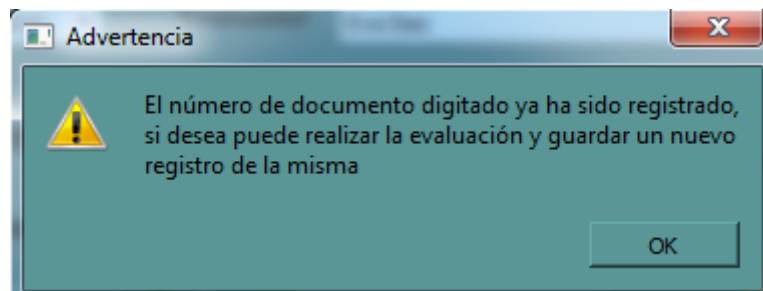


Figura 4.6. Mensaje de alerta generado por número de documento repetido.

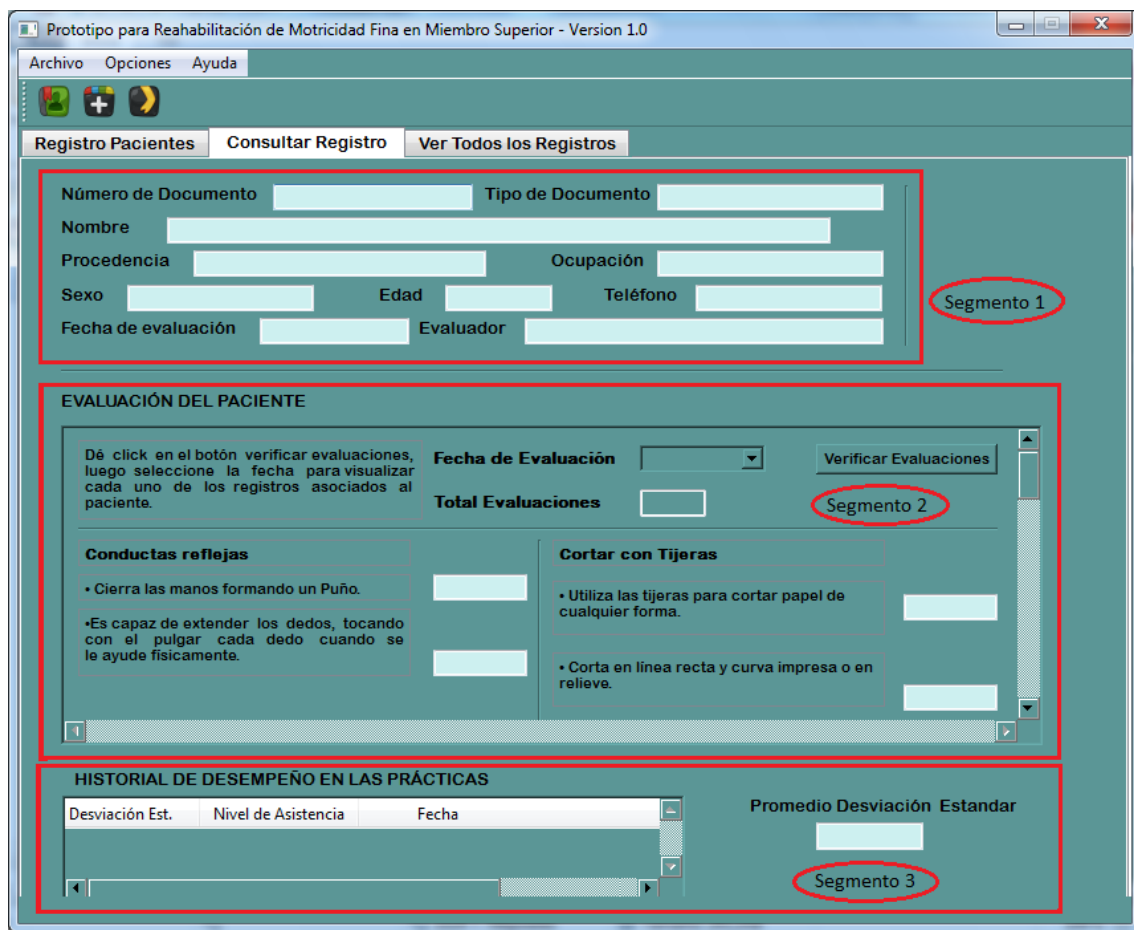


Figura 4.5. Interfaz de Usuario, Formulario dos, 'Consultar Registro'.

4.1.2 Formulario dos, 'Consultar Registro'

El formulario dos está dividido en tres segmentos, en el primero están los datos personales del paciente, en el segundo se presentan las evaluaciones asociadas a un paciente, y en la parte final está lo correspondiente al historial de desempeño del paciente. El usuario debe introducir el número de documento en el campo adecuado y dar clic en el mismo, al hacerlo se actualizará el formulario permitiendo visualizar toda la información asociada a este número. En caso de que el número introducido no esté registrado, en pantalla aparecerá el mensaje visto en la figura 4.7.

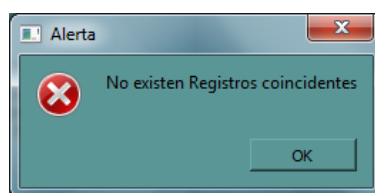


Figura 4.7. Mensaje generado por introducción de documento no registrado.

En la parte superior de los datos del segmento dos, hay un botón llamado 'Verificar Evaluaciones', al dar clic en él, se actualizarán los campos correspondientes a 'Fecha de Evaluación' y 'Total de Evaluaciones' asociados al número de documento introducido al inicio; al seleccionar alguna de las fechas de la lista desplegable se podrá visualizar la evaluación realizada en la fecha seleccionada. Es necesario utilizar la barra de desplazamiento para observar la evaluación completa (ver figura 4.8).



Figura 4.8. Botón y opción empleada para la consultar una evaluación.

Finalmente, en la parte baja de este formulario se presenta una tabla que corresponde al historial de desempeño en las prácticas de los pacientes, los datos que se pueden visualizar son: desviación estándar, nivel de asistencia, entorno de entrenamiento y fecha en la que se realizó la práctica. Al lado derecho de esta tabla está un campo destinado al promedio de desempeño del paciente.

4.1.3 Formulario tres, 'Ver Todos los Registros'

En el formulario número tres, 'Ver todos los Registros' (ver Figura 4.9), se pueden visualizar los datos personales de todos los pacientes que hayan sido guardados en la aplicación, también es posible hacer consultas por nombre de paciente o por número de documento, este formulario se hizo con el propósito de que el evaluador o usuario de la aplicación pueda ver todos los registros que hay almacenados el sistema. En caso de que no se tengan los datos necesarios para hacer la consulta unitaria en el formulario dos, podría hacer una búsqueda inicial en esta tabla, copiar el número de documento del paciente y dirigirse al formulario dos para hacer la consulta de todos los registros asociados a este paciente.

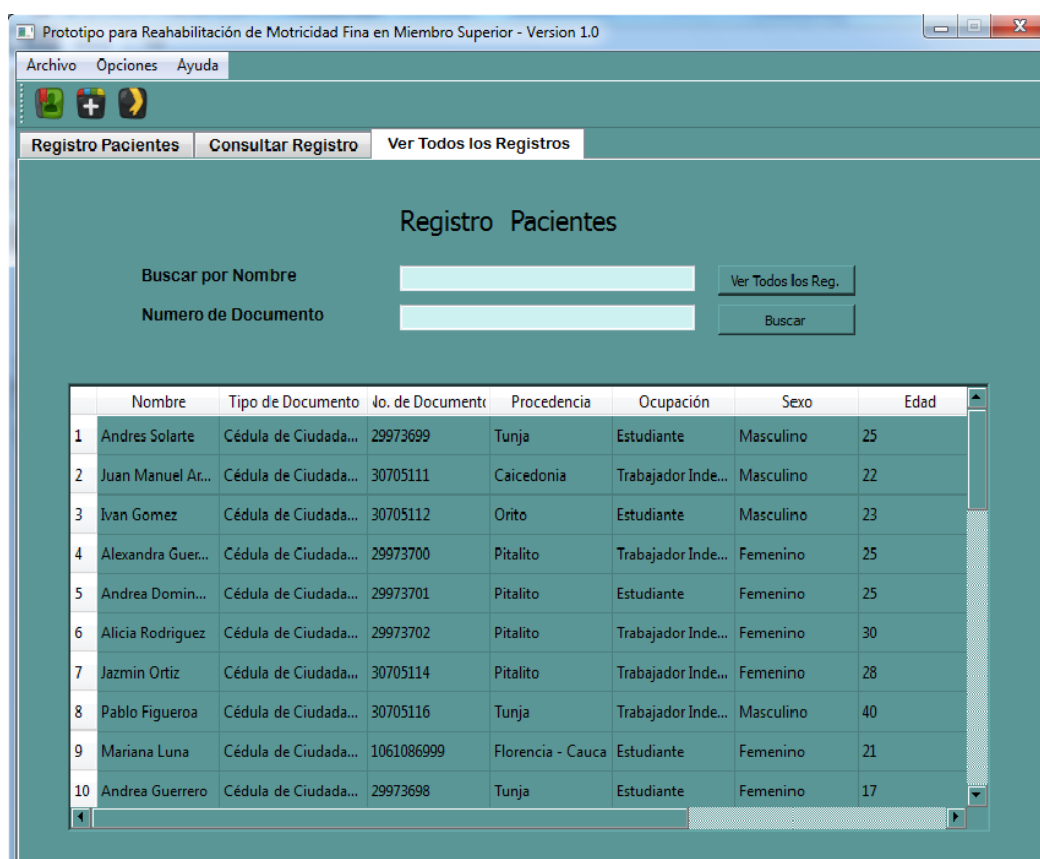


Figura 4.9. Interfaz de Usuario, Formulario tres, 'Ver todos los Registros'.

4.2 Ambientes Virtuales de Entrenamiento

El ambiente virtual de entrenamiento es la ventana de la aplicación en la que ocurre la interacción del usuario con el entorno gráfico a través del dispositivo háptico Omni Phantom. El ingreso al ambiente virtual se hace desde la interfaz de usuario mediante la pestaña de 'Opciones', en la que se despliegan las dos subdivisiones del ambiente virtual, una correspondiente al 'Seguimiento de Trayectorias' y la otra a los 'Ejercicios de Familiarización'.

En la opción de 'Seguimiento de Trayectorias', se despliegan cinco opciones más, cada una corresponde a un nivel de entrenamiento con una trayectoria diferente. En 'Ejercicios de Familiarización' se abre la opción 'Laberintos', en la que se hallan otros tipos de trayectorias pero sin asistencia de campo de fuerza; este ambiente solamente cuenta con fuerzas de realimentación para simular los contactos del cursor háptico con las trayectorias 3D cargadas en él. En la Figura 4.10, se muestra la forma de acceder a cada uno de los entornos virtuales a través de la pestaña de opciones.

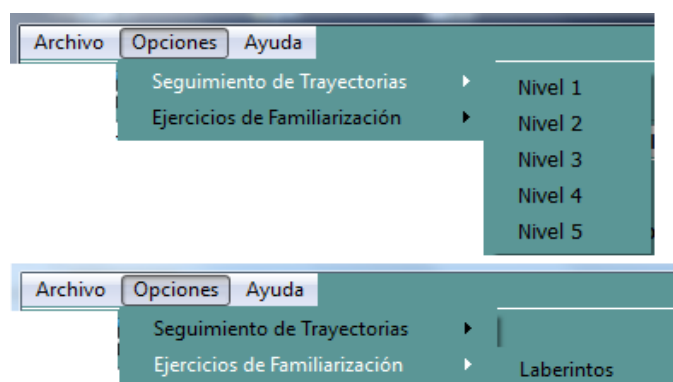


Figura 4.10. Acceso a los entornos virtuales.

4.2.1 Ejercicios de Familiarización

El objetivo de este entorno virtual de entrenamiento es que el usuario pueda probar el funcionamiento del dispositivo háptico conectado a un entorno gráfico. Al usuario se le presentan trayectorias que consisten en tuberías en las que debe pasar el cursor háptico a través de ellas; y al hacerlo podrá sentir que se encuentra rodeado por las paredes de la tubería y no se le permite retirar el cursor háptico hasta que en el entorno gráfico pueda verse que ha salido por un extremo de la misma. También en este entorno se encuentra una trayectoria similar a la utilizada en [8]; la cual tiene apariencia de un laberinto en forma de canal que puede ser recorrido con el cursor háptico haciendo presión contra la

cara inferior del mismo, y a medida que se avanza se pueden sentir las colisiones con las caras laterales. La ventana en la que se encuentran las trayectorias consta de una ficha ubicada en la parte inferior que contiene el nombre de la aplicación y los datos sus desarrolladores. En la parte superior de la ventana se encuentra el entorno virtual. La Figura 4.11, muestra la ventana de ejercicios de familiarización.

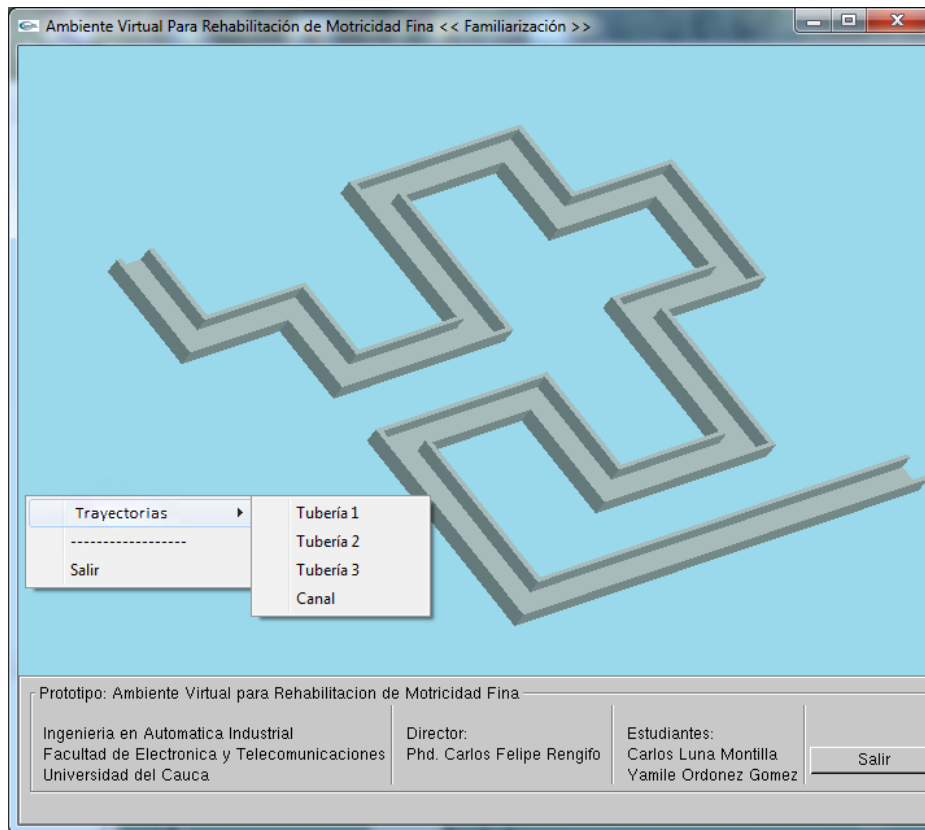


Figura 4.11. Ventana de ejercicios de familiarización.

Todas las trayectorias pueden aparecer en la misma ventana del ambiente virtual, para seleccionar la deseada se debe hacer clic derecho sobre el ambiente y aparecerán las opciones a elegir en un menú desplegable; otra manera de hacerlo es mediante el teclado, con las teclas de funciones desde F1 hasta F4. En este entorno de familiarización, se le permite al usuario girar la trayectoria con la que se encuentra practicando; esto se hace con el teclado numérico de la siguiente manera: giro sobre el eje x con la tecla 1, giro opuesto en el eje x con la tecla 2; giro sobre el eje y con la tecla 3, giro opuesto en el eje y con la tecla 4; giro sobre el eje z con la tecla 5, giro opuesto en el eje z con la tecla 6. Esto le permite al usuario elegir una perspectiva de la trayectoria con la que se sienta cómodo al momento de seguirla. Todas las trayectorias que pueden elegirse en este entorno virtual se observan en la Figura 4.12.

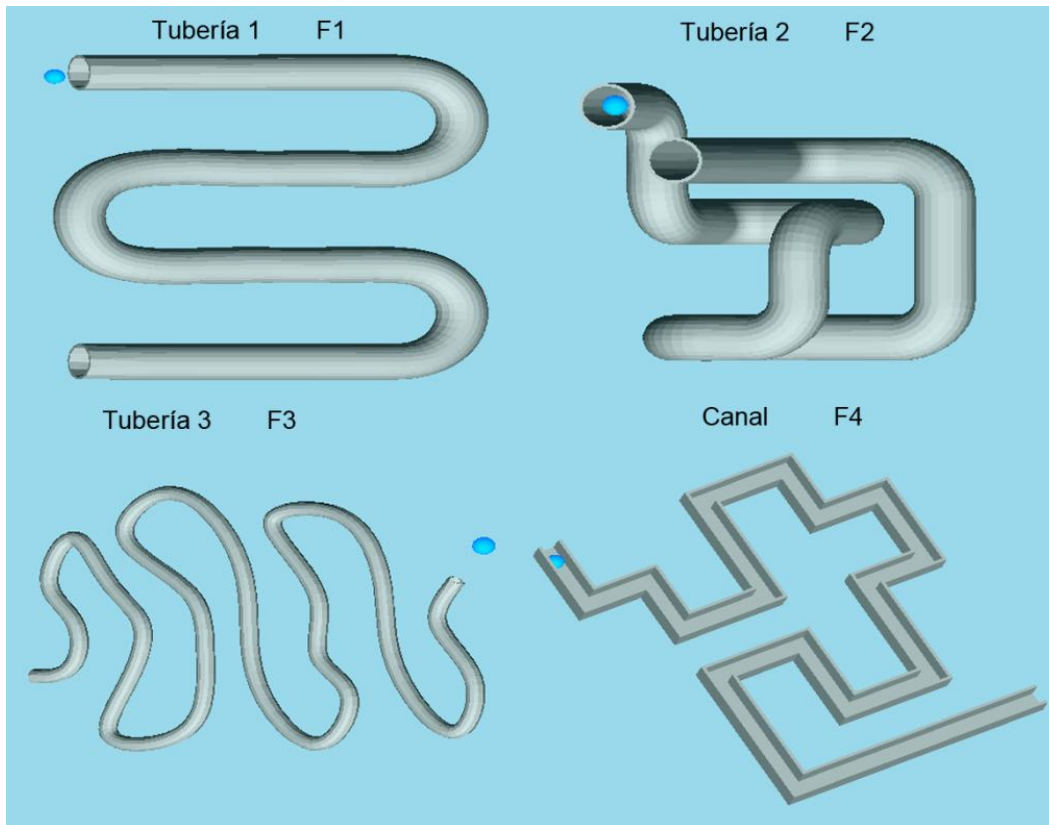


Figura 4.12. Trayectorias de ejercicios de familiarización.

Otra opción que se tiene en este entorno es la de poder mirar la forma de la trayectoria con dos tipos diferentes de primitivas de dibujado de OpenGL: las líneas y los puntos. Si se presiona la tecla F6, se observa la construcción de la trayectoria mediante la primitiva líneas y al presionar F7 con la primitiva puntos; esto puede resultar conveniente si el cursor háptico se encuentra a mitad de camino en las trayectorias tipo tubería y se desea sacarlo de ahí sin llegar a un extremo de la misma. Como la sensación de contacto no se puede sentir con las primitivas líneas o puntos, el cursor háptico queda libre en el ambiente virtual. Para restaurar la trayectoria a una forma que sea palpable, se presiona la tecla F5 y nuevamente estará dibujada con la primitiva polígonos.

El fisioterapeuta o el encargado de la supervisión de las prácticas, deberá hacer que el paciente primero entre al entorno de 'Ejercicios de familiarización', pues el manejo de dispositivos hápticos no es una práctica muy difundida en la población. La interfaz de interacción con ambientes virtuales más común es el mouse de los computadores, pero ésta solo tiene dos grados de libertad; por lo tanto pasar al manejo de una interfaz de seis grados de libertad podría resultar no muy sencillo.

4.2.2 Seguimiento de Trayectorias

Se compone de cinco entornos en los que se ha cargado una trayectoria diferente para ser recorrida con el cursor háptico mediante el apoyo de un campo de fuerza. Estas trayectorias no son cilindros huecos como en los ejercicios de familiarización, sino por el contrario son sólidas y de un diámetro reducido, el fin es representar las curvas de Bézier que fueron modelados matemáticamente en el ambiente virtual.

Recorrer estas trayectorias de aspecto cilíndrico y sólido con el cursor háptico, se logra haciendo que este último adquiera una forma toroidal; de esta manera el proxy quedará ubicado en el centro del toroide y el campo de fuerza se aplica sobre éste hacia el cuerpo de la trayectoria en cuestión, generando la sensación visual de un anillo que contiene la trayectoria en su centro mientras se dirige de un punto inicial a un punto final de la misma.

En la Figura 4.13, se muestra la ventana que contiene el ambiente virtual con la trayectoria que fue modelada mediante curvas de Bézier para el Nivel 1 de entrenamiento. Las ventanas de los cuatro niveles restantes tienen el mismo aspecto pero con una trayectoria diferente.

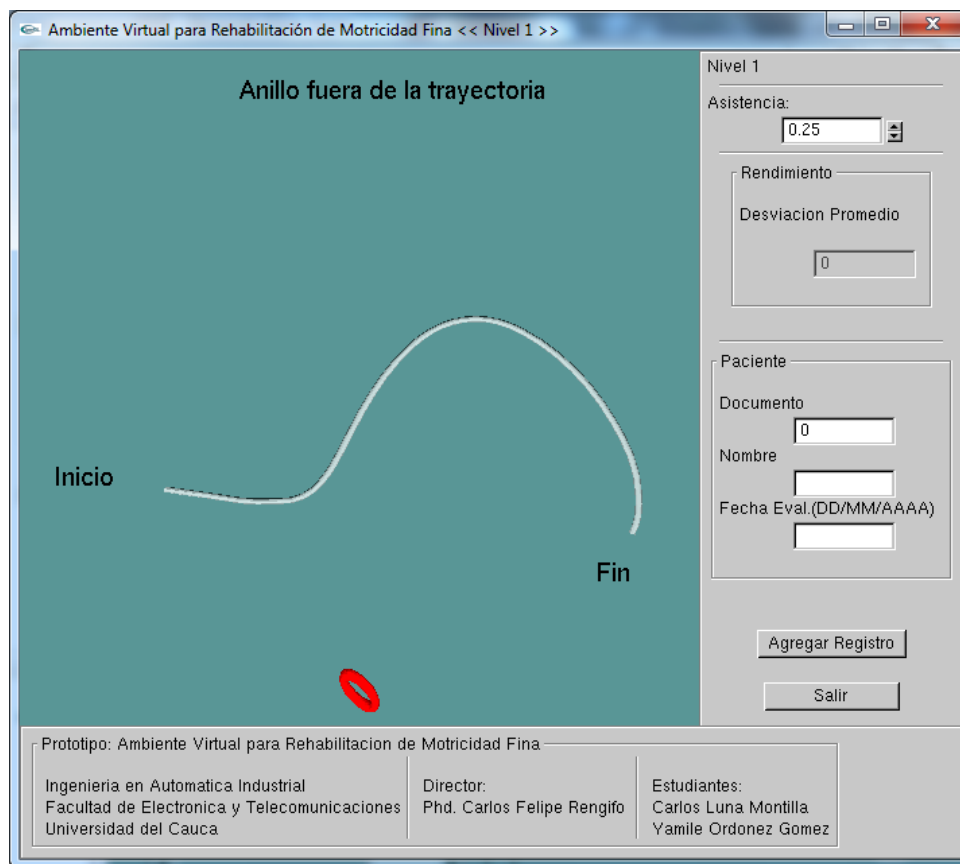


Figura 4.13. Ventana del ambiente virtual correspondiente al Nivel 1 de entrenamiento.

El área en la que se observa la trayectoria es el ambiente virtual tridimensional en el cual puede trasladarse y rotar en cualquier dirección el cursor háptico representado por un toroide. El extremo izquierdo de la trayectoria se indica como el punto de inicio y el derecho como el final; esto es importante porque el campo de fuerza se configuró para que empezara a actuar luego de que el cursor háptico entre en un determinado volumen cerca del punto de inicio, y permanece hasta que se haya alcanzado el punto indicado como fin. Si el campo de fuerza no se encuentra activado, el dispositivo háptico sólo retornará la fuerza correspondiente a la compensación de la gravedad; además el algoritmo que calcula el promedio de la desviación solamente se activa y se desactiva cuando también lo hace el campo de fuerza.

En la parte superior de la trayectoria se pueden leer unos mensajes que se activan dependiendo del comportamiento del cursor háptico en el seguimiento de la trayectoria; cada mensaje se corresponde con el color que puede tomar el toroide en una determinada ubicación. El objetivo es proporcionar señales que le permitan a la persona que realiza el seguimiento de la trayectoria percatarse de los errores que puede estar cometiendo o si está realizando la práctica correctamente.

Cuando el centro del toroide se encuentra alineado o con una desviación menor a la magnitud de su radio interno respecto a la trayectoria; el toroide se torna de color azul y aparece el mensaje “Muy bien!” en la parte superior del ambiente virtual. Cuando la desviación del centro del toroide respecto a la trayectoria es mayor a su radio interno y menor a su radio externo, el color que toma el toroide es naranja y el mensaje que aparece es “Cuidado!”. Si la desviación es mayor al radio externo del toroide, este se torna de color rojo y aparece el mensaje “Anillo fuera de la trayectoria”, lo cual puede verificarse gráficamente. Los mensajes y los colores que toma el toroide dependiendo de su posición se observan en la Figura 4.14.

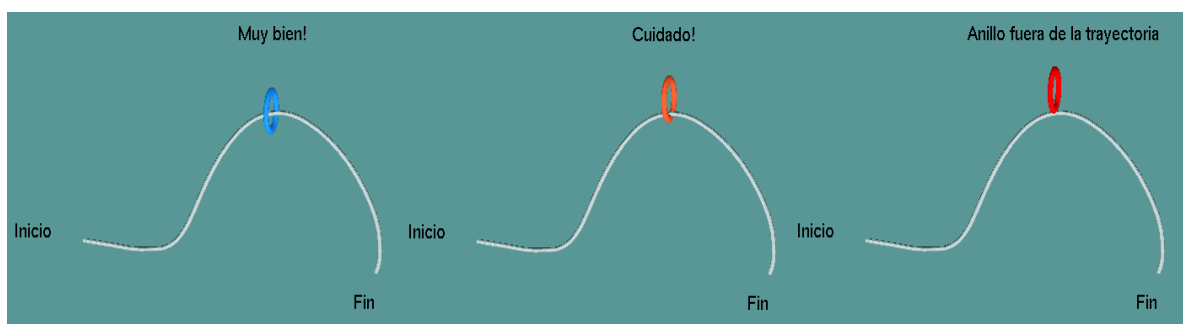


Figura 4.14. Mensajes y colores del toroide dependientes de la desviación del mismo respecto a la trayectoria.

En el lado derecho de la ventana mostrada en la Figura 4.15, se encuentra la subventana destinada a permitir la variación del grado de asistencia del campo de fuerza y a asignar el puntaje obtenido en la práctica, al registro del paciente que la realizó.

Al abrir la ventana del ambiente virtual, el fisioterapeuta encargado de supervisar la práctica debe ingresar un valor de asistencia del campo de fuerza que puede estar entre cero y uno. El valor por defecto que aparece en la casilla de asistencia es 0.25. Depende del juicio del fisioterapeuta incrementar o disminuir este valor, según lo que concluya del desempeño del paciente.

Después de configurar el grado de asistencia, se ingresa el número de documento del paciente que realizará la práctica y que debió ser registrado previamente en la interfaz de usuario; y aparecerá el nombre del paciente correspondiente. Se ingresa la fecha y entonces se procede a comenzar la práctica. A medida que el ejercicio es realizado, la casilla del promedio de la desviación se actualiza constantemente.

Cuando se termina el ejercicio, los datos correspondientes a la fecha, desviación promedio, nivel de entrenamiento y grado de asistencia se asignan al registro del paciente haciendo clic en el botón 'Agregar Registro'; después se hace clic en el botón 'Salir'. En la Figura 4.15 se muestra la subventana con todos los datos diligenciados para un paciente (ficticio) que obtuvo una desviación promedio de 12 en la realización de la práctica.

| Nivel 1 | |
|-------------------------|------------|
| Asistencia: | 0.254969 |
| Rendimiento | |
| Desviacion Promedio | 12 |
| Paciente | |
| Documento | 29973699 |
| Nombre | Andres Sol |
| Fecha Eval.(DD/MM/AAAA) | 31/10/2012 |
| Agregar Registro | |
| Salir | |

Figura 4.15. Subventana con los datos diligenciados.

Si el paciente realiza satisfactoriamente el ejercicio del Nivel 1, con una asistencia nula o muy baja del campo de fuerza, el fisioterapeuta hará que el paciente entrene en el Nivel 2, y así sucesivamente hasta el Nivel 5. Las trayectorias para los cinco niveles se observan en la Figura 4.16.



Figura 4.16. Trayectorias para los cinco niveles de entrenamiento.

Conclusiones

- Como resultado del desarrollo del presente trabajo se obtuvo la primera versión del prototipo de una Aplicación para Rehabilitación de Motricidad Fina en Miembro Superior, el cual está compuesto básicamente por una interfaz de usuario que permite el registro y administración de datos; y un ambiente virtual diseñado con el propósito de servir como medio para la realización de actividades destinadas a trabajar aspectos propios de la motricidad fina, como coordinación ojo-mano, manipulación de la mano e integración viso-motora.
- De acuerdo a los resultados obtenidos y considerando la sección 1.3.2, el ambiente virtual diseñado cumple con los requisitos de sinteticidad, tridimensionalidad, inmersión e interactividad.
- Cuando se programa el dispositivo háptico Omni Phantom con la librería HL de alto nivel, el desarrollo de una aplicación con renderizado gráfico y háptico resulta más sencillo, pero se debe ceñir a las características predefinidas que esta tiene. Para la personalización de la fuerza ejercida por el dispositivo, como lo fue la creación del campo de fuerza realizado en este trabajo; se requiere que la programación se realice a bajo nivel con la librería HD.
- El método de curvas de Bézier proporciona una forma sencilla para el modelado de trayectorias en un ambiente virtual. Para la creación de una trayectoria mediante la implementación de este método no se necesita de una gran cantidad de coordenadas para definir su curso, ni dar solución a un sistema de ecuaciones para hallar los coeficientes que definen la fórmula de una curva en particular, ni el establecimiento de tramos de trayectoria que luego se unen de forma interactiva. Todas estas características lo convierten en un método robusto y eficiente, incluso para aplicaciones que requieran acceder a cada punto que conforma la curva, en tiempo real.
- En la determinación del punto de la trayectoria que se encuentra más cercano a la posición del cursor háptico, se probaron dos métodos para minimizar la distancia y así encontrar un valor de la variable s , que evaluado en la curva de Bézier (3.8), produjera dicho punto. Estos

métodos fueron el método de Brent y un método exhaustivo. Aunque no se esperaba, resultó mucho más rápida la respuesta cuando se utiliza el método exhaustivo. El problema con la implementación del método de Brent fue el tiempo en el que el algoritmo converge hasta encontrar el valor buscado y que a veces puede converger a un punto de inflexión local y no al mínimo global de la función de distancia.

- El desarrollo de aplicaciones que incluyen dispositivos para ejercitación de miembros superiores e inferiores, deben considerar el efecto de la gravedad, para que el peso de las herramientas no causen fatiga muscular en los pacientes. En el campo de fuerza implementado en la aplicación desarrollada en este trabajo, se incluye un vector de magnitud constante destinado a contrarrestar el efecto de la gravedad, de esta manera se contribuye a la precisión del guiado sobre la trayectoria y se evita que el peso del brazo del dispositivo sea soportado por el paciente.

Trabajos futuros y recomendaciones

El avance de la tecnología de realidad virtual ha beneficiado considerablemente al área de rehabilitación motriz, colocando a su disposición elementos que entre otras cosas permiten estimular los distintos sistemas de regulación involucrados en el desempeño de las habilidades motrices. Se espera, que haciendo uso de dicha tecnología y partiendo del desarrollo de este trabajo se obtenga una nueva versión que contenga otros ejercicios, para trabajar actividades propias de la motricidad fina. Se sugiere que estos ejercicios incluyan una buena presentación gráfica, que además de la incorporación de un sistema háptico incluya sonido, con lo cual se espera mejorar el nivel de inmersión del paciente.

En la primera versión del prototipo presentado en este trabajo, solamente se considera un parámetro para la evaluación de desempeño del paciente en la realización de ejercicios de seguimiento de trayectorias. Se propone que en la nueva versión de la aplicación, se consideren más criterios; como por ejemplo, el número de colisiones del anillo con la trayectoria, la velocidad con la que el paciente hace el recorrido, la fuerza con la que impacta la trayectoria, y el temblor de la mano del paciente, para lo cual se podría considerar el estudio realizado en [8].

A lo largo de este trabajo, se expusieron algunos casos de estudio en los que se presenta evidencia científica de la efectividad de los ambientes virtuales con propósitos de rehabilitación de motricidad; sin embargo, la única manera de tener certeza que el aplicativo construido sirve a dichos fines, es la realización de un estudio clínico. Dicho estudio mostraría de manera tangible la eficacia de la aplicación. Para lo cual, se sugiere trabajar conjuntamente con el programa de Fisioterapia de la Facultad de Salud; el objetivo sería validar el prototipo Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior y establecer rangos de calificación de desempeño dependiendo de segmentación de población patológica y no patológica en diferentes rangos de edad.

Bibliografía

- [1] Francine Malouin, Carol L Richards, Bradford McFadyen, and Julien Doyon, "New perspectives of locomotor rehabilitation after stroke," *Medecine Sciences*, pp. 994,998, 2003.
- [2] Romero Tori, Claudio Kirner, and Robson Siscoutto, *Fundamentos e Tecnologia de Realidade Aumentada*. Belém-PA, 2006.
- [3] Heidi Sveistrup et al., "Experimental studies of virtual reality-delivered compared to conventional exercise programs for rehabilitation," *Cyberpsychol Behav*, pp. 243-249, 2003.
- [4] Chang-Yih Shing, Chin-Ping Fung, Tien-Yow Chuang, I-Wen Penn, and Ji-Liang Doong, "The study of auditory and haptic signals in a virtual reality-based hand rehabilitation system.," *Robotica*, pp. 211-218, 2003.
- [5] David Jack et al., "Virtual reality-enhanced stroke rehabilitation," *IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING*, pp. 308-318, 2001.
- [6] Rui Loureiro, Farshid Amirabdollahian, Susan Coote, Emma Stokes, and William Harwin, "Using Haptics Technology to Deliver Motivational Therapies in Stroke Patients: Concepts and Initial Pilot Studies".
- [7] Norali Pernaleté et al., "Eye-Hand Coordination Assessment Using a Robotic," in *International Conference on Robotics & Automation* , New Orleans, LA , 2004.
- [8] Ales Bardorfer, Marko Munih, Anton Zupan, and Alenka Primožic, "Upper Limb Motion Analysis Using Haptic Interface," *IEEE/ASME TRANSACTIONS ON MECHATRONICS*, pp. 1-9, 2001.
- [9] Roland Arsenault and Colin Ware, "Eye-Hand Co-ordination with Force Feedback," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* , New York, 2000, pp. 408-414.
- [10] Víctor Rubio J. and M Flor Basurto Zaldivar, *La evaluación curricular de la*

motricidad y la autoayuda. Madrid: Siglo XXI Editores, 1994.

- [11] Vicência Barbosa Pereira Cheib, "Avaliação do desenvolvimento motor grosseiro e de factores de risco para o atraso na aquisição destas habilidades em crianças com faixa etária de 4 a vinte e quatro meses na zona rural do distrito de Senhora do Carmo, Itabira, mg, Brasil," Universidade Federal de Minas Gerais, Belo Horizonte, Tesis Maestria 2009.
- [12] Maria Mayela Lica Aguilar, Diana Lizet Ruiz Barriga, and Adriana Patricia Gonzales Cepeda, "Relación entre ejecuciones deficientes de motricidad fina con dificultades de escritura: Análisis de un caso," *Revista de Educación y Desarrollo*, pp. 17-24, Abril-Junio 2010.
- [13] Julia Maria D'Andrea Greve, "Rehabilitação: conceito terapêutico," in *Tratado de Medicina de Rehabilitação*. Sao Paulo, Brasil: Manole, 2007, pp. 8-10.
- [14] Sergio Lianza, "Avaliação clínica da incapacidade," in *Medicina de Rehabilitação*.: Guanabara Koogan, 2007, ch. 3.
- [15] Márcia Greguol Gorgatti and Maria Tereza Silveira Bohme, "Autenticidade científica de um teste de agilidade para indivíduos em cadeira de rodas," *Revista Paulinista de Educação Física*, no. 17, pp. 41-50, junio 2003.
- [16] Jannet Carr and Roberts Shepherd, *A motor relearning programme for stroke*, Segunda edición ed.: Oxford: Butterworth Heinemann, 1987.
- [17] Grigore Burdea, "Key Note Address: Virtual Rehabilitation-Benefits and Challenges," *CAIP Center, Rutgers University*, 2002.
- [18] Maureen K. Holden, "Virtual Environments for Motor Rehabilitation: Review," *CYBERPSYCHOLOGY & BEHAVIOR*, vol. 8, no. 3, pp. 187-211, Junio 2005.
- [19] Melina de Vasconcelos Alberio, "Um Ambiente Virtual Colaborativo para Treinamento Cirúrgico," Instituto Militar de Engenharia, Rio de Janeiro, Tesis de Maestria 2005.
- [20] Claudio Kirner and Tereza Kirner, "Virtual Reality and Augmented Reality Applied to Simulation Visualization," *Simulation and Modeling: Current Technologies and Applications*, pp. 391-419, 2008.
- [21] Grigore C. Burdea and Philippe Coiffet, *Virtual Reality Technology*.: John Wiley & Sons, 2003.

- [22] "Programmer's Guide Open Haptics Toolkit Version 2.0," SensAble Technologies, Woburn, Manual de Usuario 2005.
- [23] Cagatay Basdogan and Mandayam A. Srinivasan, "Haptic Rendering in Virtual Environments," in *IEEE International Conference on Robotics*, 2002.
- [24] Helber Mayorca Torres and Adrinia Lucia Torres Silva, Entorno gráfico de un simulador quirúrgico 3D, 2011, Tesis de Pregrado.
- [25] Daniel Ricardo Ramos Tovar and Luis Eduardo Camayo Gómez, Diseño y simulación de una mano robot con actuadores de nitinol para rehabilitación, 2012, Tesis de Pregrado.
- [26] L. M. García Raffi, M. J. Pérez Peñalver, E. A. Sánchez Pérez, and M. Figueres Moreno, *Matemática Asistida por Ordenador, MAO*, 2001, Editorial Universidad Politécnica de Valencia.
- [27] W. Thomas and Sederberg, *An Introduction to Bézier Curves*, 2003.
- [28] Renate Kempf and Jed Hartman, *OpenGL on Silicon Graphics Systems*, 2005.
- [29] Santiago Hermira Anchuelo, *Introducción a OpenGL y sus primitivas*, Guía laboratorio.
- [30] Anónimo, *Aprenda Qt4 desde hoy mismo*, 2010, Versión de distribución libre.
- [31] Oscar Andrés Vivas Albán, *Introducción al Qt y al Qt Creator*, 2011, Guía de Clase.
- [32] Francisco Javier Melero Rus, *Desarrollo de Aplicaciones con Qt*, 2007, Escuela Técnica Superior de Ingeniería Informática.
- [33] Elena Muñoz España, *Tratamiento y Visualización de Imágenes 3D*, 2011, Guía.
- [34] Daniel Molina Cabrera, *Tutorial CMake*, 2009, Tutorial CMake.
- [35] SensAble Technologies, *Phantom Omni User's Guide*, 2003-2004.

Anexo A

Presentación de herramientas complementarias

En esta sección se exponen las herramientas distintas a las ofrecidas por OpenHaptics Toolkit de Sensable Technologies, que fueron empleadas para la realización del prototipo de la Aplicación de Rehabilitación de Motricidad fina en Miembro Superior.

Se empleó OpenGL como motor de renderizado gráfico, Qt, para el diseño de la interfaz de usuario, SQLite, para almacenar y administrar los datos de la aplicación, y CMake, para generar el proyecto de la aplicación de interfaz de usuario (aplicación Qt), para que esta funcione sobre la plataforma de Visual Studio 2008. A continuación se aborda superficialmente cada una de estas herramientas.

OpenGL

OpenGL (Open Graphics Language) es una aplicación utilizada para crear primitivas graficas 2D y 3D, disponible en la mayoría de las plataformas, con soporte a varios tipos diferentes de hardware. Puede ser utilizada en cualquier aplicación gráfica, por ejemplo en juegos, en modelamiento e incluso hasta CAD. Fue inicialmente desarrollado por la SGI (SiliconGraphics, Inc) como una plataforma para varios propósitos gráficos [28].

Las ventajas principales ofrecidas por esta API son:

- Pertener al grupo de estándares gráficos abiertos, independientes de la plataforma.
- Hay control con las versiones anteriores para que las aplicaciones funcionen.
- Las aplicaciones OpenGL son consistentes en cualquier hardware compatible con su API.
- Nuevos hardware con su API son compatibles.
- Funcionan en cualquier tipo de máquinas, desde electrodomésticos hasta supercomputadores.
- No es necesario crear funciones para un hardware específico.

- Existen varios libros y documentos con códigos accesibles y de fácil entendimiento.

Completa información respecto a OpenGL, puede ser encontrada en su página oficial, o en [28], [29].

Qt

Qt es un framework multiplataforma, hoy en día es muy empleado por la facilidad con la que permite crear interfaces de usuario (GUI o GraphicsUser Interface); por ser multiplataforma, las aplicaciones escritas utilizando Qt pueden ser compiladas en distintas plataformas, como Linux, Windows y Mac.

Qt comenzó a ser desarrollado en 1991 en Noruega por Harvard Nord y Eirik Chamb-Eng, en una empresa llamada Quasar Technologies. Tres años después, ellos fundaron la empresa TrollTech que pasaría después a llamarse Trolltech [30].

En la actualidad Qt cuenta con una licencia dual, por un lado la comercial que es paga y por otro la LGPL que es gratuita, la licencia comercial permite el desarrollo de aplicaciones propietarias con restricciones de licencia.

Qt fue desarrollado con el objetivo inicial de facilitar tareas a los programadores de C++. Pese a que Qt usa bibliotecas estándares de C++, el programador no las usa de un modo convencional, lo que en realidad hace es emplear bibliotecas propias de Qt, que a través de un pre-procesador (denominado compilador Meta Object o MOC) facilita el desarrollo de aplicaciones.

Qt ofrece a los programadores una serie de herramientas y documentación que permiten que la realización de aplicaciones sea sencilla. Entre estas herramientas están:

- QtDesigner, que permite crear diálogos e interfaces gráficamente.
- QtLinguist, esta herramienta permite crear aplicaciones multilingües.
- QtAssistant, generador de documentación
- Qmake, generador de Makefiles multiplataforma.

La estructura del framework en mención se puede observar en la Figura A.1.

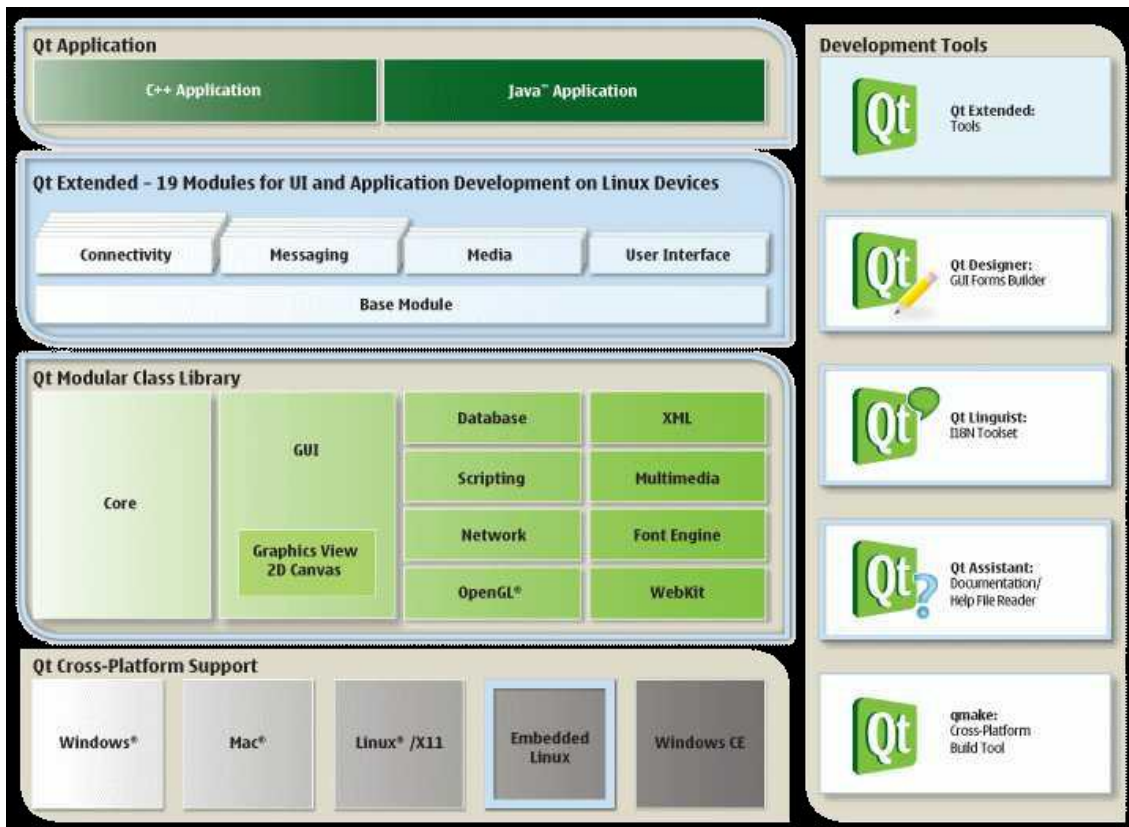


Figura A.1. Estructura librería Qt⁸.

La interfaz de usuario diseñada en este trabajo fue realizada con QtCreator, este es un IDE que emplea librerías de Qt, y que tiene como principales características la programación visual y programación dirigida por eventos.

La programación visual permite al usuario diseñar el aspecto gráfico de la aplicación mediante el uso y distribución de controles, también llamados widgets, estos podrían ser: formularios, botones, menús, cuadros de texto entre otros; la interacción entre los mismos se puede realizar a través de eventos.

Entre tanto en la programación dirigida por eventos, el usuario escribe el código que se ejecutará en respuesta a determinados eventos, llamados también *slots*, por ejemplo: pulsar un botón y cerrar una ventana o elegir una opción en un menú. La labor del programador se centra en asociar a cada evento el comportamiento adecuado.

En términos generales las ventanas son clases, los controles o widgets son clases, y los eventos o slots son métodos de las ventanas. Una ventana creada

⁸ <http://www.linuxfordevices.com/c/a/News/Trolltech-and-Qtopia-revd-and-renamed/>

por el usuario es una subclase de la clase ventana, propia del IDE, esta podría ser de tipo QMainWindow, QDialog o QWidget.

Otra característica que se resalta en QtCreator es que este presenta las prestaciones propias de un IDE avanzado, entre las que están: una sintaxis coloreada, compleción automática de código, ayuda sensible al contexto, inspector de objetos, diseñador visual, compilador y depurador integrado.

Completa información al respecto es dispuesta en la página oficial de este IDE, algunas guías básicas para la instalación y creación de aplicaciones con QtCreator pueden ser encontradas en [30], [31] y [32].

CMAKE

El CMake es una herramienta libre, multiplataforma, disponible en su sitio oficial; es utilizada para configurar y administrar el proceso de construcción de aplicaciones. Los ficheros independientes llamados CMakeLists.txt se usan para definir el proceso de construcción y establecer las dependencias [33].

Al ejecutar CMake se generan los ficheros necesarios, dependiendo del compilador y sistema operativo que se esté empleando. En este trabajo se emplea CMake para compilar la aplicación diseñada en QtCreator, para que esta pueda ser usada desde Visual Studio 2008, en donde se recopilan todas las librerías requeridas en el diseño del prototipo expuesto en este trabajo.

Guías de instalación y uso de CMake están disponibles en [34], [33] y [24].

SQLITE

Es un sistema de bases de datos relacionales, RDBMS, (Relational Database Management System) de dominio público, entre las principales tareas que puede desempeñar están:

- Permitir el almacenamiento de datos al sistema.
- Permitir recuperar los datos y trabajar con ellos.
- Proveer herramientas para capturar, editar y manipular los datos.
- Permitir crear reporte e informes con los datos.

SQLite incluye una herramienta que es básicamente una línea de comandos llamada SQLite3, denominada en Windows sqlite3.exe, que permite al usuario introducir y ejecutar comandos SQL contra una base de datos SQLite.

En el desarrollo de este trabajo se empleó SQLiteExpert, la cual es una potente herramienta gráfica que permite administrar bases de datos SQLite3; además, permite visualizar de forma clara como está funcionando dicha base de datos.

SQLiteExpert, cuenta con una interfaz gráfica de usuario, que contribuye a que el uso de la misma resulte casi intuitivo; una de las razones por las que se escogió este RDBMS es porque QtCreator, el IDE en el que se diseñó de interfaz de usuario cuenta con un driver denominado 'QSQLITE', que le permite realizar la conexión directa entre la aplicación y la base de datos.

Anexo B

Modelado de las trayectorias en un programa CAD y su adecuación para ser importadas mediante OpenGL.

El motor de renderizado gráfico OpenGL, permite cargar en un ambiente virtual modelos realizados en un programa de Diseño Asistido por Computadora (CAD), siempre que estos modelos se hayan transformado a un formato que OpenGL pueda soportar; este formato es el Wavefront 3D Object.

Un archivo desarrollado como Wavefront 3D Object, tiene la extensión .obj y pertenece a la categoría de archivo de imagen en 3D desarrollado por Wavefront Technologies⁹.

El programa CAD utilizado para la creación del modelo tridimensional fue Solid Edge. Que en su entorno de creación “Conjunto”, Figura B.1, permite el uso de una aplicación que facilita la generación de tuberías o trayectorias cilíndricas basándose en un perfil de dibujado, denominada “XpresRoute”, Figura B.2.



Figura B.1. Selección del entorno “Conjunto”, es Solid Edge.

⁹ www.fileinfo.com/extension/obj

Un boceto en Solid Edge es una operación que se utiliza en todos los entornos de creación de piezas para realizar un dibujo libremente. Se crea un boceto en el plano xy , en el que se inserta una imagen .bmp correspondiente a la gráfica que fue generada en Matlab, Figura B.3.

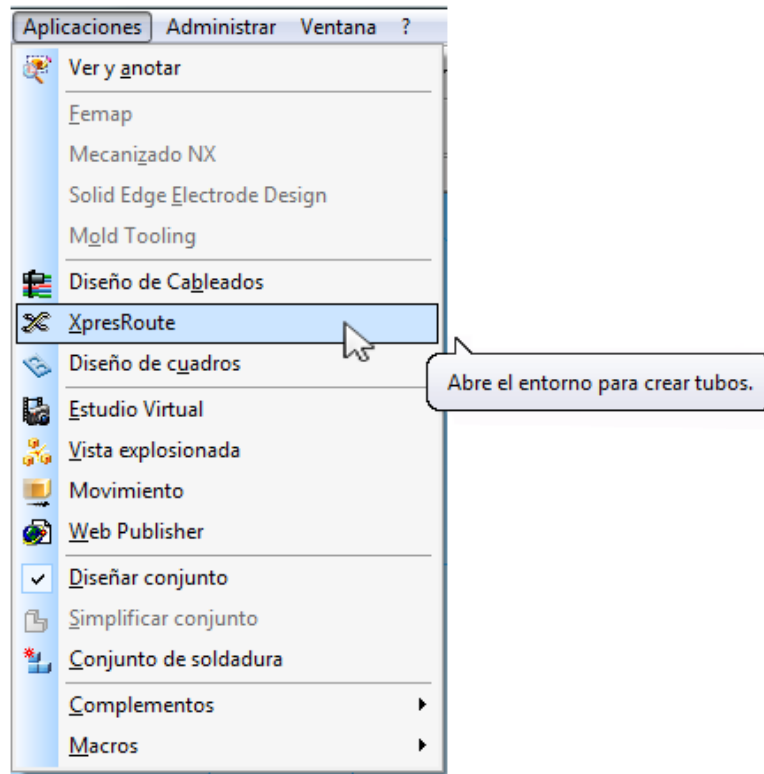


Figura B.2. Selección de la aplicación para crear tuberías.

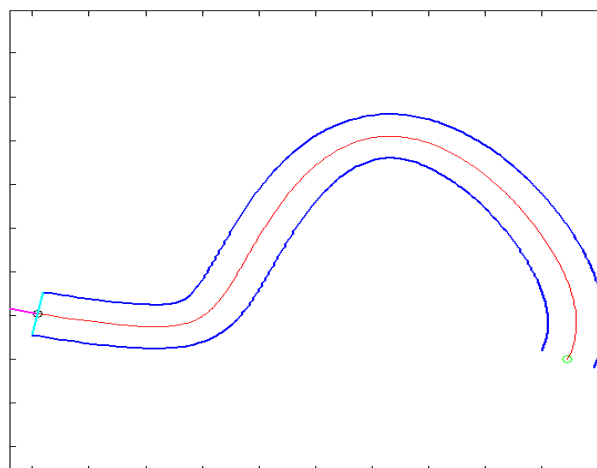


Figura B.3. Gráfica de la curva de Bézier para el nivel 1 en Matlab.

Sobre el boceto que contiene la imagen, se crea uno nuevo para calcar la línea central de la trayectoria. Este último es el que se utiliza como perfil de dibujado

de la tubería. El resultado final de la creación de una trayectoria en 3D se puede ver en la Figura B.4.

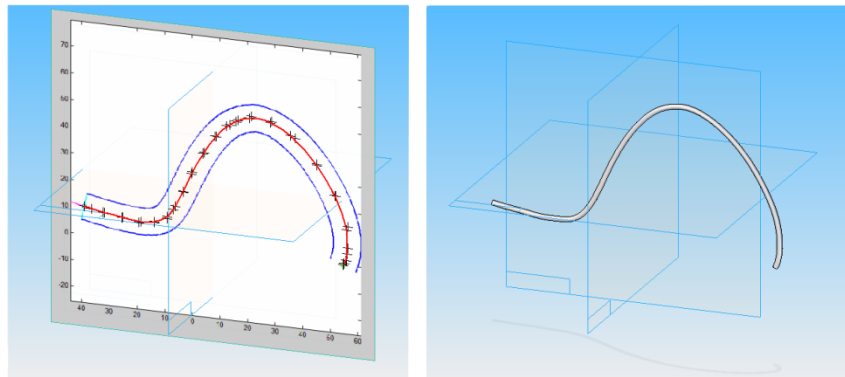


Figura B.4. Izquierda, boceto perfil de dibujado de la herramienta XpresRoute. Derecha, trayectoria en 3D implementada en Solid Edge.

La trayectoria que se observa en la Figura B.4 Derecha, se guarda con la extensión .stl, que es un formato de archivos para estereolitografía (Stereolithographic File) que pueden ser leídos por múltiples programas CAD¹⁰. El archivo .stl es importado a Blender, programa de modelado de gráficos tridimensionales, en el que se hace la conversión final de .stl a .obj.

Una muestra del ambiente de trabajo de Blender en el que se puede ver la trayectoria del Nivel 1 en proceso de conversión se observa en la Figura B.5.

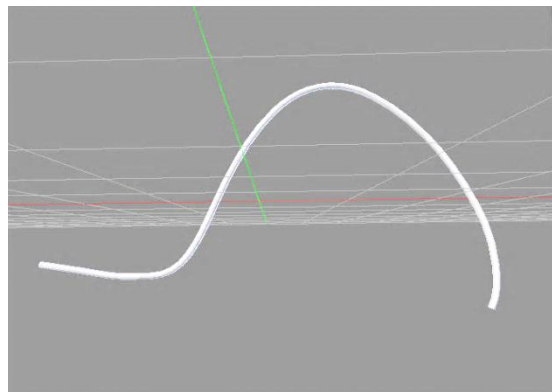


Figura B.5. Ambiente de trabajo del programa Blender.

Luego de cambiar el formato de la figura 3D a Wavefront 3D Object, se puede por medio de OpenGL cargar al ambiente virtual dentro del espacio o lienzo definido para la renderización gráfica, en el cual se pueden realizar las funciones de esta librería; afuera de él no hay visualización de objetos 3D.

¹⁰ www.fileinfo.com/extension/stl

Anexo C

Instalación del prototipo Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior.

La aplicación desarrollada cuenta con un instalador que fue creado con el objetivo de volverla más portable. 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 IEEE 1394 de 6 pines.

Las características del computador pueden influir en la calidad del renderizado háptico y gráfico.

Antes de realizar la instalación de la aplicación, se requiere que se instalen los drivers del dispositivo Phantom Omni, para que este pueda funcionar correctamente.

A continuación se muestran los pasos a seguir en la instalación del prototipo Ambiente Virtual para Rehabilitación de Motricidad Fina en Miembro Superior.

Paso 1:

Dentro de la carpeta “Instalador Ambiente Virtual para RMF”, se encuentra otra con el nombre “Release”, al interior de la cual se halla el instalador llamado “SetupAmbienteVirtual”. Sobre este último se da doble clic para iniciar la instalación, Figura C.1.

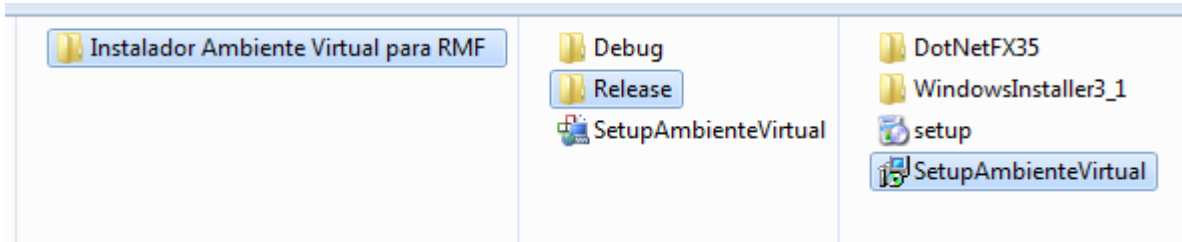


Figura C.1. Ubicación del instalador.

Paso 2:

Cuando se inicia la instalación, aparece un cuadro de diálogo que se muestra en la Figura C.2. Si se desea continuar con el proceso de instalación se oprime en el botón “siguiente”, de lo contrario se oprime “cancelar”.

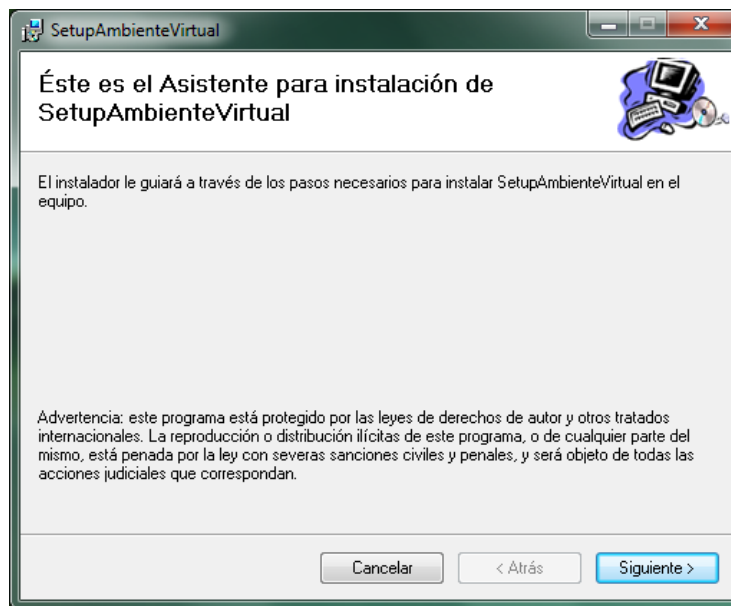


Figura C.2. Cuadro de diálogo inicial de instalación.

Paso 3:

Continuando con la instalación de la aplicación, el siguiente paso es seleccionar la carpeta en la que se guardarán los archivos. La dirección por defecto podría ser cambiada desde el botón “Examinar” o editando la cadena manualmente. Ver Figura C.3.

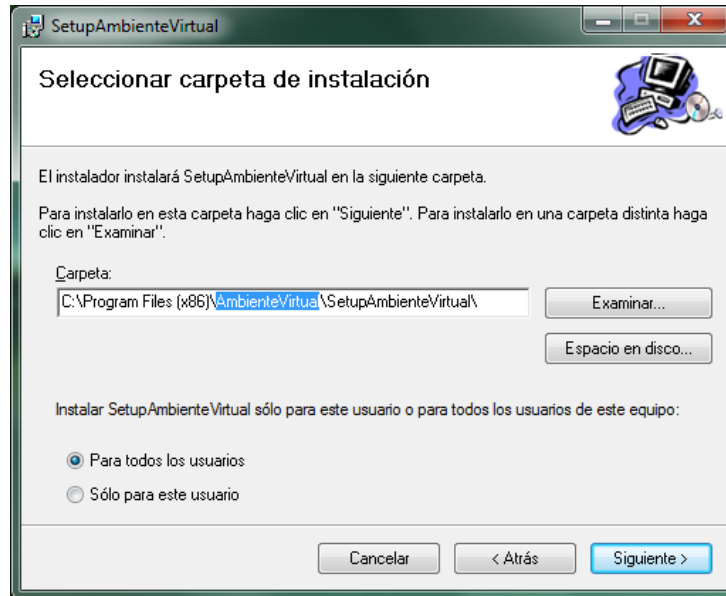


Figura C.3. Selección de ubicación de archivos.

Paso 4:

Después de dar clic en el botón “Siguiente” en la ventana mostrada en la Figura C.3, se confirma la instalación y se espera a que el proceso tenga lugar. Una vez se haya completado la instalación, se cierra el cuadro de diálogo.

El ejecutable, es almacenado en la carpeta que se seleccionó en el paso 3, con el nombre de “APLICACIÓN.exe”, del cual se debe crear un acceso directo en el escritorio haciendo clic derecho sobre éste; y se selecciona la opción enviar al escritorio.

Si después de ejecutar la aplicación desde el acceso directo en el escritorio se presentan problemas en el almacenamiento de los registros de pacientes; se debe seguramente a los permisos de administrador que tiene la carpeta en la cual se realizó la instalación.

Para solucionar este inconveniente, se debe conceder todos los permisos a la carpeta que contiene el ejecutable. Esta acción se realiza mediante la opción propiedades, en la pestaña ‘seguridad’, haciendo clic en el botón ‘Editar’, como se muestra en la Figura C.4.

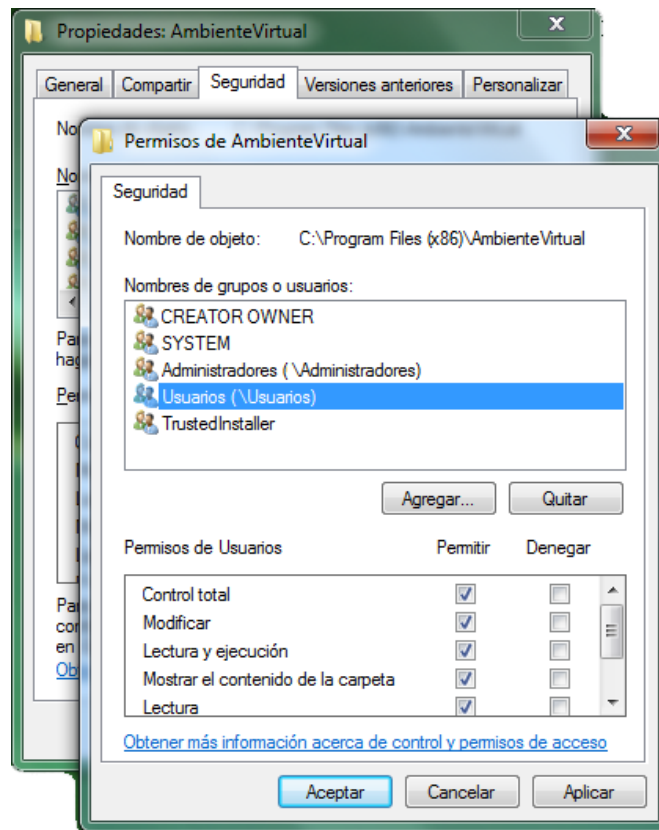


Figura C.4. Edición de los permisos de la carpeta que contiene el ejecutable de la aplicación.