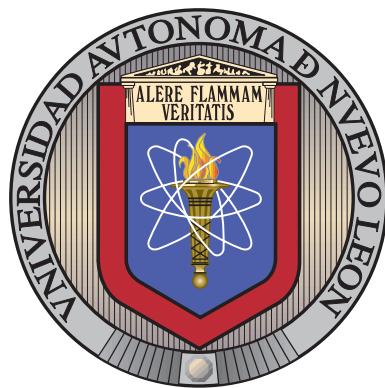


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE LICENCIATURA



APLICACIONES DE REALIDAD VIRTUAL EN SISTEMAS
DE ENTRENAMIENTO PARA TAREAS FÍSICAS

POR

GERARDO SAÚL GAUSIN VALLE

EN OPCIÓN AL GRADO DE

INGENIERO EN TECNOLOGÍA DE SOFTWARE

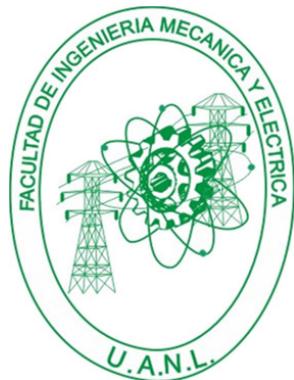
SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

NOVIEMBRE 2014

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

DIVISIÓN DE ESTUDIOS DE LICENCIATURA



APLICACIONES DE REALIDAD VIRTUAL EN SISTEMAS
DE ENTRENAMIENTO PARA TAREAS FÍSICAS

POR

GERARDO SAÚL GAUSIN VALLE

EN OPCIÓN AL GRADO DE

INGENIERO EN TECNOLOGÍA DE SOFTWARE

SAN NICOLÁS DE LOS GARZA, NUEVO LEÓN

NOVIEMBRE 2014

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

División de Estudios de Licenciatura

Los miembros del Comité de Tesis recomendamos que la Tesis «Aplicaciones de realidad virtual en sistemas de entrenamiento para tareas físicas», realizada por el alumno Gerardo Saúl Gausin Valle, con número de matrícula 1450272, sea aceptada para su defensa como opción al grado de Ingeniero en Tecnología de Software.

El Comité de Tesis

Dra. Satu Elisa Schaeffer

Asesor

Dr. Luis Martín Torres Treviño

Revisor

M.C. Carlos Bernardo Garza Treviño

Revisor

Vo. Bo.

M.C. Arnulfo Treviño Cubero

División de Estudios de Licenciatura

San Nicolás de los Garza, Nuevo León, noviembre 2014

AGRADECIMIENTOS

Agradezco a la Dra. Elisa Schaeffer por asesorarme cada semana, motivarme y alentarme a continuar avanzando con este trabajo, ya que sin ello no hubiera podido llegar a tener un avance considerable. Agradezco a mis padres por seguir apoyándome en la carrera sin importar los altos y bajos financieros dentro del hogar y el día del año.

De igual forma agradezco al Dr. Luis Martín Torres Treviño y al M.C. Carlos Bernardo Garza Treviño por haber tomado parte de su tiempo para realizar revisiones a este presente trabajo.

Se agradece a los participantes por su apoyo a la evaluación de la usabilidad.

RESUMEN

Gerardo Saúl Gausin Valle.

Candidato para el grado de Ingeniero en Tecnología de Software.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio:

APLICACIONES DE REALIDAD VIRTUAL EN SISTEMAS DE ENTRENAMIENTO PARA TAREAS FÍSICAS

Número de páginas: 114.

OBJETIVOS Y MÉTODO DE ESTUDIO: El objetivo general es realizar una simulación de un entrenamiento por medio de la realidad virtual, asimismo tomando retroalimentación de las acciones corporales del usuario que irán siendo requeridas según las instrucciones del entrenamiento, además de tomar el tiempo de desempeño.

Los objetivos específicos son crear una base de desarrollo para ayudar a entrenar a personas para que puedan controlar una situación de peligro por medio de una serie de instrucciones que son presentadas dentro de la simulación, crear un sistema que

ayude a evaluar el tiempo tomado y acciones, reducir el espacio utilizado contra una situación real y crear una interacción lo más natural posible.

La metodología se compone por una revisión de literatura existente sobre trabajos similares a este trabajo, después se realiza una exploración de tecnologías existentes tomando en cuenta sus características para luego realizar un diseño de un prototipo tomando como referencia las tecnologías existentes, después se implementa el prototipo al que posteriormente se realizan evaluaciones de funcionalidad y usabilidad.

CONTRIBUCIONES Y CONCLUSIONES: Se creó un sistema de realidad virtual el cual da la facilidad al usuario de realizar movimientos dentro del mundo virtual dadas las acciones realizadas en el mundo real como levantar mano para interactuar con las instrucciones que aparecen a lo largo de las actividades realizadas por el usuario dentro del sistema de realidad virtual. Por medio de las evaluaciones de usabilidad se identificaron aspectos que requieren de cambios y mejora en el sistema de realidad virtual. También se realizaron evaluaciones de tiempo de respuesta del sistema de realidad virtual los cuales ayudaron a identificar los posibles problemas que ocasionaron una alta latencia de los movimientos que realiza el usuario en la mundo real al tomar efecto en el mundo virtual.

Firma del asesor: _____

Dra. Satu Elisa Schaeffer

ÍNDICE GENERAL

Resumen	v
Índice general	vii
Índice de figuras	xii
Índice de cuadros	xiii
1. Introducción	1
1.1. Hipótesis	2
1.2. Objetivos	2
1.3. Estructura	3
2. Antecedentes	5
2.1. Realidad virtual	6
2.1.1. Mundo virtual	7
2.1.2. Inmersión	8
2.1.3. Retroalimentación sensorial	9

2.1.4. Interactividad	9
2.1.5. Ambiente de colaboración	10
2.1.6. Elementos de realidad virtual	10
2.1.7. Diferencia entre mundo virtual y ambiente virtual	11
2.2. Herramientas	12
2.2.1. Desarrollo en software	12
2.2.2. Desarrollo en hardware	15
3. Literatura existente	22
3.1. Trabajos existentes	22
3.2. Estudio comparativo	26
3.3. Área de oportunidad	28
4. Solución propuesta	29
4.1. Metodología de desarrollo de software	30
4.2. Fases de desarrollo	31
4.3. Exploración de tecnologías	32
4.3.1. Mundo virtual	33
4.3.2. Inmersión	34
4.3.3. Retroalimentación sensorial	35
4.3.4. Interactividad	38

4.4. Planeación de implementación	38
4.5. Diseño de prototipo	40
4.6. Implementación	48
5. Experimentos	62
5.1. Diseño	62
5.1.1. Protocolo de pensamiento en voz alta	63
5.1.2. Análisis de latencia	64
5.1.3. Evaluación heurística	65
5.2. Resultados	71
5.2.1. Protocolo de pensamiento en voz alta	71
5.2.2. Análisis de latencia	74
5.2.3. Evaluación heurística	80
6. Conclusiones	84
Bibliografía	88
Apéndice	90
A. Historia del usuario del sistema de realidad virtual	91
B. Diagramas funcionales del sistema de realidad virtual	93

C. Instrucciones de la actividad del sistema de realidad virtual	98
D. Pruebas de bondad de ajuste de las latencias	100
E. Resultados de la evaluación heurística	104

ÍNDICE DE FIGURAS

2.1.	Partes del Kinect.	17
2.2.	Partes del Oculus Rift.	19
2.3.	Ejemplo de imagen con y sin distorsión de <i>pincushion</i>	21
4.1.	Diagrama de flujo general del sistema de realidad virtual.	39
4.2.	Vista áerea del entorno virtual.	47
4.3.	Herramienta de Unity en el que se manipulan los modelos 3D.	49
4.4.	Escena de prueba de comunicación con cliente UDP en Unity.	50
4.5.	Prueba de comunicación con cliente UDP en Unity.	50
4.6.	Ejemplo de imagen de profundidad tomada desde el Kinect.	51
4.7.	Objeto tipo terreno en Unity.	52
4.8.	Implementación y manipulación de paredes para limitar al usuario. .	52
4.9.	Implementación de los planos y colores correspondientes.	53
4.10.	Modelo del mundo virtual terminado.	53
4.11.	Posiciones fáciles y difíciles para la detección de las partes del cuerpo.	54

4.12. Detección de la orientación de usuario.	56
4.13. Se detecta el usuario caminando.	57
4.14. Estados del mundo virtual del sistema de realidad virtual.	61
5.1. Porcentajes de latencias totales por fase.	76
5.2. Fases de las latencias del sistema de realidad virtual.	77
5.3. Análisis de los valores intermedios de la latencia en fase B.	78
5.4. Análisis de los valores altos de la latencia en fase B.	79
5.5. Análisis de los valores bajos de la latencia en fase B.	79
B.1. Diagrama de estados máquina del mundo virtual.	94
B.2. Diagrama de secuencias del mundo virtual (parte 1).	95
B.3. Diagrama de secuencias del mundo virtual (parte 2).	96
B.4. Diagrama de flujo de la aplicación que obtiene información del usuario.	97
D.1. Ajuste de los valores de la fase A a la distribución de Poisson.	101
D.2. Ajuste de los valores de la fase B bajos a la distribución de Poisson. .	102
D.3. Ajuste de los valores de la fase C a la distribución exponencial.	103

ÍNDICE DE CUADROS

3.1. Características de desarrollo en software.	28
3.2. Características de herramientas en hardware.	28
4.1. Claves de realidad virtual y tecnologías a explorar.	33
4.2. Herramientas de desarrollo de entornos en 3D.	34
4.3. Dispositivos para ver entornos virtuales.	36
4.4. Características de dispositivo	37
4.5. Datos identificados de la aplicación del mundo virtual.	43
4.6. Datos del usuario que son procesados e identificados.	46
E.1. Resultado de la evaluación de las heurísticas.	104
E.2. Clasificación de los problemas encontrados.	109

CAPÍTULO 1

INTRODUCCIÓN

En la actualidad realizar un entrenamiento hacia un grupo de personas puede llegar a ser difícil debido al costo y tiempo que se invierten, además sin contar con otros factores como el peligro, costo de los materiales a utilizar y el tiempo de vida de los materiales. El costo de un entrenamiento puede llegar a incrementar dependiendo de la cantidad de personas que se desea entrenar y el tiempo con el que se plantea dar un entrenamiento. Al dar un entrenamiento a un grupo de personas se tiene como problema la gestión del tiempo debido a que cada individuo tiene diferentes ritmos de aprendizaje. Haciendo claro que cada individuo se le tiene que dar cierta atención para que pueda llegar a aprender la técnica o método. En algunos casos ciertos entrenamientos requieren de realizar simulacros que en ocasiones son costosos debido al área en el que son aplicados y el material requerido para llevarlo a cabo. En ciertos entrenamientos se requiere utilizar un material en específico para realizar técnicas o métodos en el que se les involucra, además de tomar en cuenta que algunos materiales tienen un tiempo vida de uso limitado.

Lo anterior es un conjunto de problemas típicos que se enfrenta un organismo para alguien que desea llevar a cabo un entrenamiento. Por medio de la realidad virtual se pueden evitar este conjunto de problemáticas; el costo de un entrenamiento se basa simplemente en el equipo necesario para implementar el sistema de realidad virtual y en donde el tiempo que se invierte depende del progreso del individuo,

mientras que factores como peligro pueden ser evitados e incluso pueden ser simulados a niveles que nunca se han visto en la vida real, esto ayudando a que el individuo a entrenar tenga una mejor capacitación a diferencia de un entrenamiento en la vida real. Los materiales no tienen costo: se crea un objeto virtual que puede ser usado dentro del mundo virtual, dando como resultado la eliminación de los tiempo de vida de los materiales. En este trabajo se plantea realizar un sistema de realidad virtual base para realizar este tipo de aplicaciones. Dando como ventajas que el entorno de entrenamiento puede ser cambiado a la necesidad del tipo de entrenamiento deseado, recibir retroalimentación del entrenamiento (movimientos de brazos, piernas, hombros, postura de cabeza, etcétera) y el poder realizar operaciones que son peligrosas en el mundo real dentro de un mundo virtual.

1.1 HIPÓTESIS

La *hipótesis* planteada es la siguiente: la realización de entrenamiento en un sistema de realidad virtual de forma eficiente y cómoda es viable a través de la evaluación de las acciones corporales de las personas que utilicen dicho sistema.

1.2 OBJETIVOS

El *objetivo general* es realizar una simulación de un entrenamiento por medio de la realidad virtual, asimismo tomando retroalimentación de las acciones corporales del usuario que irán siendo requeridas según las instrucciones del entrenamiento, además de evaluar las acciones tomadas y el tiempo de desempeño.

Los *objetivos específicos* planteados sobre este trabajo son los siguientes:

- Ayudar a entrenar a personas para que puedan responder a una situación por

medio de una serie de instrucciones que son presentadas dentro del sistema realidad virtual.

- Crear un sistema que ayude a evaluar el tiempo tomado y acciones, reducir el espacio utilizado contra la situación real.
- Crear una interacción lo más natural posible y crear un escenario para ser usado en la simulación tomando en cuenta una actividad a realizar.

1.3 ESTRUCTURA

Este presente trabajo se compone de siete capítulos. En el capítulo de introducción se presenta la hipótesis, el objetivo general y los objetivos específicos del proyecto. Luego en el capítulo de antecedentes se presenta la teoría básica para llegar a tener un buen entendimiento del tema. Después en el capítulo de literatura existente se presentan trabajos documentados existentes y se comparan las soluciones propuestas por ellos por medio de un estudio comparativo para encontrar las áreas de oportunidad.

En el capítulo de solución propuesta se presentan los requerimientos y el procedimiento de la realización del proyecto.

En el capítulo de experimentos dentro de la sección de diseño se presentan tres experimentos que ayudan a evaluar la usabilidad y latencia de la solución propuesta. Después en sección de evaluación se evalúan los experimentos propuestos y se obtienen resultados en los que se reportan y visualizan los resultados para identificar el comportamiento del sistema de realidad virtual.

Por último en la capítulo de conclusiones se presentan las conclusiones y oportunidades a futuro del proyecto, así como las conclusiones de los resultados de las

evaluaciones de los experimentos del sistema de realidad virtual.

CAPÍTULO 2

ANTECEDENTES

En este capítulo se ve una introducción sobre el concepto de realidad virtual según Sherman y Craig [14] y los elementos que lo componen. Luego se presentan herramientas de desarrollo tanto como en software y hardware.

Dentro de las herramientas de desarrollo en software se involucran fundamentos de programación utilizando como lenguajes de programación JavaScript¹, Java² y C#³, y como ambientes de desarrollo Processing⁴, MonoDevelop⁵ y Unity⁶. Se explica también el funcionamiento del protocolo de transporte, debido a que es utilizado para establecer comunicación de una aplicación a otra.

En las herramientas de desarrollo en hardware se involucran los dispositivos Kinect y Oculus Rift. Por último se mencionan los conceptos principales para entender el desarrollo del mundo virtual y detección del movimiento corporal del usuario.

Los conceptos de álgebra que se necesitaron para tener un entendimiento del proceso de la obtención del movimiento y posición del usuario son temas sobre vectores, rotación, translación y matrices. Lengyel [6] habla sobre puntos importantes de la álgebra para entender la creación y manipulación de gráficos 3D.

¹Sitio no oficial de JavaScript http://www.w3schools.com/js/js_intro.asp.

²Sitio oficial de Java <http://www.oracle.com/technetwork/java/index.html>.

³Sitio oficial de C# <http://msdn.microsoft.com/en-us/vstudio/hh341490.aspx>.

⁴Sitio oficial de Processing <http://processing.org>.

⁵Sitio oficial de MonoDevelop <http://monodevelop.com>.

⁶Sitio oficial de Unity <http://unity3d.com/unity>.

2.1 REALIDAD VIRTUAL

En esta sección se discute el concepto de realidad virtual en donde se explica el origen y significado del mismo, asimismo también se explican los conceptos de los elementos que definen la realidad virtual haciendo referencia a Sherman y Craig [14].

En tiempos prehistóricos se han transmitido ideas y experiencias por medio de pinturas rupestres, como por ejemplo las experiencias de caza, el cómo realizar la siembra o el cómo se realiza un sacrificio para los dioses. Dichas pinturas eran utilizadas para transmitir la experiencia de la historia contada en ella. Se dice que estos fueron los primeros indicios de la realidad virtual y que al paso de la tecnología este concepto fue evolucionando debido a que los humanos queremos expresar cada vez mejor nuestras ideas por cada nuevo medio posible.

La realidad virtual ha sido uno de los nuevos medios para transmitir ideas y experiencias. Esta tecnología está tomando un gran impacto en estos últimos años gracias al avance de la tecnología e ideas que han desarrollado personas con grandes talentos. Una buena forma de entender el concepto de realidad virtual es definiendo qué es realidad y qué es virtual.

La Real Academia Española⁷ define realidad como “existencia real y efectiva de algo”; siguiendo esta definición se entiende que la realidad es algo que existe y es real, que tiene alguna propiedad física. Por otro lado, la Real Academia define virtual como “que tiene existencia aparente y no real” dicha definición da a entender que la virtual existe pero no puede tener una propiedad física. La definición en conjunto de los dos conceptos de realidad y virtual sería lo siguiente “existencia real y efectiva de algo que tiene existencia aparente y no real”, es contraproducente la definición que es algo que es real pero no es real. Esto se debe a que en la realidad virtual la

⁷El diccionario de la Real Academia Española se encuentra disponible en <http://www.rae.es/>.

realidad puede ser simulada e incluso la sensación del tacto dando una sensación de realidad.

Según Sherman y Craig [14], la manera simple de explicar realidad virtual es “un lugar que existe y que podemos experimentar”. También comentan que la experiencia de la realidad virtual se compone de cuatro elementos un mundo virtual, inmersión, retroalimentación sensorial (respondiendo a las respuestas de entrada del usuario) e interactividad.

Sherman y Craig [14] definen cuatro elementos que componen la realidad virtual: mundo virtual, inmersión, retroalimentación sensorial e interactividad. A continuación se discuten estos cuatro elementos, su definición y su propósito, además se ven otros conceptos como ambientes de colaboración y la diferencia entre el concepto de mundo virtual y ambiente virtual.

2.1.1 MUNDO VIRTUAL

Ahora se explica el concepto del mundo virtual y cuál es su rol dentro del concepto de realidad virtual. El *mundo virtual* es el contenido donde se comparten las ideas y experiencias. Puede existir incluso sin ser mostrado por un sistema de realidad virtual (conjunto de hardware y software que producen la experiencia de realidad virtual). El mundo virtual es como guión de una película en el que se sigue una secuencia dentro de un escenario con iluminación y sonido de ambiente. Cuando se simula un guión, se ve en un mundo virtual, se hace una inmersión física e interactúa con los objetos dentro de ese mundo virtual, se dice que se experimenta la realidad virtual.

Según Sherman y Craig [14], un mundo virtual cuenta con dos definiciones, en donde la primera lo define como un mundo imaginario que mayormente se manifiesta

a través de un medio, el segundo lo define como una descripción de un conjunto de objetos en un espacio los cuales están siendo administrados por un conjunto de reglas y relaciones.

2.1.2 INMERSIÓN

En el contexto de realidad virtual, la inmersión es una alternativa a la realidad o un punto de vista. Según Sherman y Craig [14] el mundo es percibido de dos maneras diferentes: se puede percibir un mundo alternativo o el mundo normal desde otro punto de vista. En el mundo alternativo éste se puede representar de un espacio que existe en cualquier parte, o puede ser producto de pura imaginación. Un ejemplo de mundos alternativos son los productos de compositores, artistas, novelistas entre otros relacionados a desarrollo creativo. El poder de la imaginación puede crear una poderosa inmersión con los productos creativos, sin embargo, estos productos solamente están diseñados para realizar una interacción unidireccional de autor a la audiencia, el punto de vista siempre es en tercera persona, el diálogo y la historia está predefinida.

En la realidad virtual el efecto de entrar a un mundo virtual comienza por la inmersión física que de la inmersión mental. La inmersión puede ser usado de dos maneras: inmersión mental e inmersión física (sensorial); en donde siendo inmersión se refiere a estar en cualquiera de los dos estados sintiendo la experiencia del mismo. Sherman y Craig [14] comentan que la comunidad de realidad virtual ha seleccionado la palabra presencia para representar este concepto de inmersión, y que su significado lo definen como “siendo mentalmente inmerso”.

2.1.3 RETROALIMENTACIÓN SENSORIAL

La realidad virtual permite a sus participantes⁸ la libertad de posicionar su cuerpo, de tal forma que cualquier movimiento puede realizar eventos dentro del mundo virtual. También es el medio por el cual se pueden experimentar realidades imaginadas con sentidos físicos. Además permite realizar escenarios no posibles en el mundo real, reduciendo el peligro de un daño físico que puede ser ocasionado en un escenario del mundo real. La retroalimentación sensorial es la parte esencial de la realidad virtual.

El sistema de realidad virtual para obtener la retroalimentación sensorial debe de poder realizar un seguimiento del movimiento corporal del participante. Un sistema típico de realidad virtual realiza seguimiento de la posición de la cabeza del participante y al menos una mano o en su defecto por un dispositivo de control; existen sistemas avanzados que detectan y hacen seguimiento del movimiento de todas las articulaciones del cuerpo. En la actualidad existe una gran variedad de tecnologías para cumplir con el seguimiento del movimiento de los participantes.

2.1.4 INTERACTIVIDAD

Dentro de la realidad virtual, la *interactividad* es el acto de responder a acciones que realiza el participante dentro del mundo virtual. Un ejemplo de interactividad son los videojuegos y simulaciones por computadora (tal como simulaciones de clima, simulaciones de vuelo entre otros), ya que responden dependiendo del valor dado de entrada. Hay que tener en consideración que para que exista interactividad no es necesario realizarla por medio de gráficas computacionales. En la realidad virtual la interactividad es asociado con la habilidad de que el participante se mueva físicamente.

⁸En la literatura de realidad virtual a los usuarios que utilizan el sistema de realidad virtual se les llama participantes.

te con el mundo virtual, así como tomando ventaja del seguimiento del movimiento de la cabeza para poder tener distintos puntos de vista dentro del mundo virtual.

2.1.5 AMBIENTE DE COLABORACIÓN

El *ambiente de colaboración* es una extensión del elemento de interacción y refiere a que múltiples usuarios pueden interactuar dentro de un mismo espacio o mundo virtual, dentro del los usuarios pueden percibir otros usuarios permitiendo una interacción mutua. La representación del usuario dentro del mundo virtual o simulación es referida como *avatar*.

El ambiente de colaboración es una interacción entre múltiples usuarios dentro de un mundo virtual; no es necesariamente manifestado en la realidad virtual. Un ambiente de realidad virtual colaborativo puede ser referido como multipresencia o múltiparticipante. Esto es una característica importante para múltiples aplicaciones de realidad virtual, incluyendo esos de simulación de combate y entrenamiento, y la industria de videojuegos de realidad virtual, los cuales pueden envolver un equipo de jugadores humanos con computadoras. Es claro que la realidad virtual tiene muchos usos como lo es la telepresencia; un ejemplo de él lo podría ser realizar cirugías en Japón estando en Cuba por medio de un sistema de realidad virtual. Sin embargo aquí entran muchos factores como la calidad de la conexión a internet, el cual podría ser controlado realizando un sofisticado sistema que prevenga ese tipo de problemática.

2.1.6 ELEMENTOS DE REALIDAD VIRTUAL

Sherman y Craig [14] comentan que al combinar los cuatro elementos se puede decir que la realidad virtual es “un medio compuesto de simulaciones de computadora interactivas que obtienen la posición y acciones del participante, y dando

como respuesta una o más acciones dentro del mundo virtual dando como resultado la sensación de siendo mentalmente inmerso en la simulación (el mundo virtual)”. Un sistema de realidad virtual puede tener una gran variedad de dispositivos que pueden ser para mantener el seguimiento de la posición del usuario, seguimiento de las articulaciones del cuerpo, sensores electromagnéticos musculares, sensores electromagnéticos de encefalograma, entre otros.

Uno de los principales dispositivos que son típicamente usados para dar la inmersión física al mundo virtual es un casco o también conocido como pantalla montada en la cabeza (HMD que por sus siglas en inglés, *head-mounted display*). Este dispositivo muestra una pantalla para cada ojo dando cada uno un distinto punto de vista al desplegar gráficas computacionales para poder dar una sensación de inmersión además de tener la posibilidad de ajustarse según el tamaño de la cabeza. Dentro de este dispositivo se tiene un conjunto de sensores que le comunican a la computadora la posición de la cabeza para saber cuáles gráficas computacionales mostrar, de esta forma el participante podrá ver el mundo virtual tal como lo haría en el mundo real. Adicionalmente se le pueden agregar dispositivos como reconocimiento de voz, guantes de detección de flexibilidad o sensores electromagnéticos para enriquecer la interacción.

2.1.7 DIFERENCIA ENTRE MUNDO VIRTUAL Y AMBIENTE VIRTUAL

Actualmente existen dos definiciones que se utilizan mucho y que son susceptibles a la confusión: mundo virtual y ambiente virtual. Estos dos términos son utilizados como sinónimos de realidad virtual, sin embargo el uso de ambiente virtual era utilizado anteriormente para definir realidad virtual; dicha palabra puede ser definida como mundo virtual o realidad virtual. A mediados de los ochentas, in-

vestigadores de la NASA⁹ usaban frecuentemente la expresión “ambiente virtual” para describir su trabajo en crear una interfaz que permitiera que una persona experimente una escena creada a computadora desde el punto de vista de la primera persona, el cual describe lo que actualmente es realidad virtual.

2.2 HERRAMIENTAS

En esta sección se describen las herramientas que son utilizadas en software y hardware. En las herramientas de software se discuten los ambientes de desarrollo que son utilizados con sus respectivos lenguajes de programación. En las herramientas de hardware se presentan en detalle los dispositivos Kinect y Oculus Rift.

2.2.1 DESARROLLO EN SOFTWARE

Las herramientas de desarrollo en software seleccionadas fueron debido a las funcionalidades que ofrecen, mismas que son requeridas para cumplir con las necesidades del diseño del prototipo. Las herramientas de desarrollo en software son llamadas por la comunidad de desarrolladores como ambientes de desarrollo (IDE por sus siglas en inglés, *interface development environment*), estos ofrecen un sistema de administración de funcionalidades para que desarrolladores puedan entender de una forma organizada las funcionalidades del lenguaje de programación de su preferencia o librería¹⁰ utilizada. A continuación se presentan los lenguajes de programación y los ambientes de desarrollo utilizados, esto de una manera general después se discuten cada uno de ellos a detalle.

Los lenguajes de programación que son utilizados en estos ambientes de desarrollo son: Processing, Javascript y C#. Los ambientes de desarrollo que son utiliza-

⁹Sitio de la NASA <http://www.nasa.gov>.

¹⁰Software preconstruido que ahorra tiempo e implementación de funcionalidades específicas.

dos son: Processing, Monodevelop y Unity. Processing además de ser un lenguaje de programación también es un ambiente de desarrollo el cual promociona el desarrollo de software por medio de arte visual. Aún cuando su diseño inicial fue para el desarrollo del arte y para enseñar fundamentos de programación, éste ha ido evolucionando de tal forma que se ha convertido en una herramienta de uso profesional.

Al utilizar el ambiente de desarrollo Processing se eligieron para desarrollar las librerías UDP y SimpleOpenNi. La capa de transporte es una de las capas en el modelo TCP/IP y está diseñada para permitir que las entidades iguales en los clientes de origen y destino puedan llevar a cabo una conversación. Actualmente se cuentan con dos protocolos de comunicación el TCP (protocolo de control de transmisión) y UDP (protocolo de datagrama de usuario) [18].

TCP es un protocolo confiable y orientado a la conexión, que permite que un flujo de bytes que se origina en una máquina se entregue sin errores en cualquier otra máquina en la interred¹¹. TCP divide el flujo de bytes entrantes en mensajes discretos y pasa cada uno de ellos a la capa de interred. El proceso que hace TCP al llegar al destino es reensamblar en el flujo de salida los mensajes recibidos. Además TCP maneja el control de flujo para asegurarse de que un emisor rápido no sature a un receptor lento con más mensajes de los que puede manejar.

UDP es un protocolo no confiable y no orientado a la conexión para las aplicaciones que no desean una comunicación de secuenciación o el control de flujo de TCP y que desean proporcionar el suyo. También tiene un amplio uso de consultas únicas de peticiones de servicio de tipo cliente-servidor en un solo envío, así como aplicaciones en las que la entrega puntual es más importante que la precisa, como en la transmisión de voz o vídeo.

¹¹Se le llama interred a un conjunto de redes interconectadas.

En Processing, la librería de UDP¹² que se proporciona permite la comunicación de UDP, así como el soporte de multidifusión de mensajes. Se seleccionó el protocolo UDP debido a que en el sistema de realidad virtual lo importante es mantener una latencia baja, por lo cual UDP es una buena opción para establecer comunicación rápida entre las aplicaciones a comunicar.

La librería SimpleOpenNi¹³ provee acceso a toda la información del Kinect, los cuales son requeridos para realizar tareas complejas. Además de proveer herramientas y ayuda por medio de la comunidad de desarrolladores. Se eligió esta librería debido a que los desarrolladores iniciales que mantienen esta librería fueron unos de los desarrolladores del Kinect en la compañía de PrimeSense — posteriormente comprado [5] por Apple (<http://www.apple.com>) — en colaboración con Microsoft (<http://www.microsoft.com>). Además de cubrir las necesidades requeridas como lo son detección de las articulaciones y partes del cuerpo del participante detectado.

MonoDevelop es una herramienta multiplataforma diseñada principalmente para el uso del lenguaje de programación C#, y otros lenguajes de programación relacionados con .NET¹⁴. Esta herramienta proporciona a los desarrolladores una manera rápida de escribir aplicaciones web y de escritorio en Linux¹⁵, Windows¹⁶ y Mac OS X¹⁷. Una de sus ventajas principales de esta herramienta es su característica de portar aplicaciones .NET creadas en Windows a Linux y Mac OS X manteniendo un solo código base para todas las plataformas.

Unity es un ecosistema de desarrollo de videojuegos en tres dimensiones (3D). Las variables utilizadas para la representación de un modelo 3D son comúnmente *x*, *y* y *z*, los cuales se utilizan para representar su ancho, altura y profundidad.

¹²Sitio de la librería de UDP en Processing <http://ubaa.net/shared/processing/udp>.

¹³Sitio de SimpleOpenNi en Processing <https://code.google.com/p/simple-openni>.

¹⁴Sitio oficial de .NET <http://www.microsoft.com/net>.

¹⁵Sitio oficial del Kernel de Linux <https://www.kernel.org>.

¹⁶Sitio oficial de Windows <http://windows.microsoft.com/en-us/windows/home>.

¹⁷Sitio oficial de Mac OS X <https://www.apple.com/mx/osx>.

Unity tiene un motor de renderizado de gran alcance integrado con un conjunto de herramientas intuitivas y flujos de trabajo que son utilizados para crear modelos 3D interactivos, así como modelos en 2D. Unity cuenta con una tienda de activos (por su nombre en inglés, *asset store*) en donde se puede encontrar gran cantidad de materiales para crear proyectos enriquecidos, además de contar con una comunidad de desarrolladores activos en donde responden preguntas sobre cualquier problema o duda al desarrollar en Unity.

2.2.2 DESARROLLO EN HARDWARE

Para crear un sistema de realidad virtual se necesita tener hardware que pueda transmitir el mundo virtual hacia el participante como por ejemplo una pantalla que muestre el mundo virtual hacia el participante, esto también aplica para obtener la posición y acciones del participante. Para ello se realizó una investigación sobre el hardware que podría cumplir con las características del diseño del prototipo. Los dispositivos utilizados son Kinect y Oculus Rift.

KINECT

El Kinect es un dispositivo que contiene una cámara de profundidad que detecta la distancia de los objetos que están enfrente de él. Esto lo hace por medio de luz infrarroja para crear imagen de profundidad el cual captura en donde los objetos han sido puestos. El uso de imágenes de profundidad ayuda a evitar que la computadora realice mucho trabajo computacional para identificar la distancia de los objetos por medio de una imagen a color. Normalmente para identificar personas u objetos a nivel computacional se tiene que comenzar a identificar los píxeles [15] por el cual están siendo representados. Identificar personas u objetos por medio de píxeles tiene sus desventajas como la variabilidad de color del píxel debido a la intensidad de la

luz por la que ha sido capturada la imagen, falta de luminosidad entre otros.

Por un lado, en una imagen de profundidad cada píxel dice la distancia que hay entre la cámara de profundidad y el objeto u objetos que se encuentran enfrente de él, y gracias a ello se puede identificar fácilmente las dimensiones de los objetos detectados, así como sus distancias. La imagen de profundidad almacena información tridimensional de los objetos frente a la cámara de profundidad, de tal forma que es posible construir objetos en modelos tridimensionales que han sido capturados por la cámara de profundidad.

Debido a la cámara de profundidad del Kinect es posible el uso del mismo para la detección y seguimiento de personas y al mismo tiempo localizar cada una de sus articulaciones y partes del cuerpo. Cabe de resaltar que Microsoft aprovecha el Kinect principalmente para partes el cuerpo de las personas y sus movimientos.

El Kinect es un producto de Microsoft, sin embargo Microsoft no creó el Kinect. El Kinect es un producto de muchos años de investigación académica por la división de investigación de Microsoft y la comunidad de visión computación. El hardware del Kinect fue desarrollado por una empresa llamada PrimeSense, la cual es una empresa de origen israelí y que ha producido anteriormente otras cámaras de profundidad utilizando la misma técnica de proyección infrarroja. PrimeSense trabajo de manera cercana con Microsoft para producir cámaras de profundidad que funcionen con el software y algoritmos desarrollados por la división de investigación de Microsoft. En esencia, PrimeSense licenció el diseño del hardware a Microsoft para crear su propio producto a la medida.

En noviembre 2010 el Kinect fue lanzado a la venta por Microsoft argumentando que esta tecnología va revolucionar la manera en la que se juega videojuegos. En mayo 2013 Microsoft anunció el Kinect 2.0 en donde señalan que incluirá detección de hasta seis personas de manera simultánea y con una resolución de 1080 píxeles



Figura 2.1 – Partes del Kinect.

de las cámaras.

El Kinect está compuesto de un proyector infrarrojo, una cámara RGB [15] (cámara a color), cámara infrarroja y micrófonos, tal como se muestra en la figura 2.1. Como se había mencionado anteriormente el proyector infrarrojo proyecta hacia los objetos que están enfrente de la cámara mientras que la cámara infrarroja obtiene la información que tiene de los objetos gracias al proyector.

Oculus Rift

Oculus Rift es una pantalla montada en la cabeza el cual tiene como fin de que entusiastas, desarrolladores de videojuegos, investigadores, entre otros puedan realizar aplicaciones de realidad virtual experimentando la inmersión y presencia dentro del mundo virtual. El Oculus Rift fue desarrollado por la compañía de Oculus VR y que gracias a una campaña de recaudación en Kickstarter¹⁸ lograron desarrollar el producto. Actualmente en el mercado existe un solo modelo de Oculus Rift que está disponible para el público en general el cual es llamado paquete de desarrollo 1 (Development Kit Version 1) o DKV1. Cabe resaltar que Oculus RV anunció en marzo del 2014 el nuevo paquete de desarrollo 2 o DKV2, agregando mejoras de

¹⁸Kickstarter es un sitio para obtener fondos para el desarrollo de una idea <https://www.kickstarter.com/>.

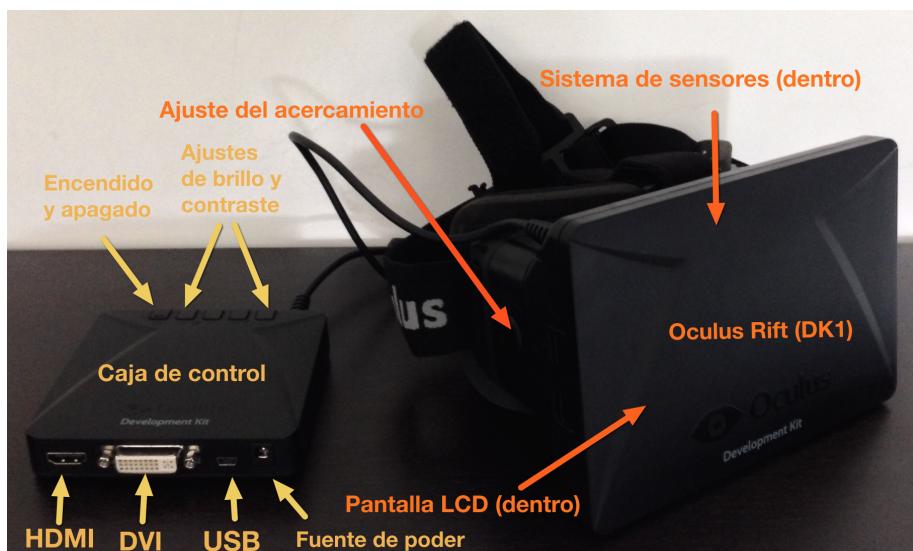
latencia y resolución de imagen. El DKV2 se encuentra a la venta al público desde septiembre del 2014.

El Oculus Rift tiene un precio de 300 dólares, cuenta con una resolución de 640×800 píxeles, tiene un campo de visión horizontal de más de 90 grados (tres veces el campo de visión horizontal humano), contiene una pantalla de siete pulgadas en donde se despliega el mundo virtual y tiene un sistema de sensores para obtener datos del movimiento de la cabeza. También cuenta con una carcasa de plástico: en él se contiene una pantalla LCD y un sistema de sensores que toma datos de la aceleración, velocidad de rotación, fuerza de campos electromagnéticos y dirección; este sistema de sensores cuenta con una velocidad de frecuencia de mil veces por segundo. Afuera de la carcasa se encuentra un cable que va directo a la caja de control del Oculus Rift; en él se envía información del sistema de sensores y recibe información que va a ser desplegada en la pantalla LCD. La pantalla que incorpora Oculus Rift está dividida entre dos partes, de tal forma que en cada ojo se ve una parte distinta en donde cada ojo tiene una resolución de 640×800 píxeles. Oculus Rift también cuenta con un mecanismo para ajustarlo por la parte superior de la cabeza y detrás de la cabeza. Cuenta con una perilla de ajuste para alejar o acercar la pantalla LCD de los ojos. Además de que cuenta con lentes intercambiables según la visión de la persona que lo vaya a usar. En la figura 2.2(a) se muestran algunas partes del Oculus Rift de la parte posterior, en donde se observan los distintos ajustes con los que se cuenta.

En la caja de control del Oculus Rift se tiene un cable que entra directamente al Oculus Rift el cual recibe la información del sistema de sensores y envía información a desplegar en la pantalla LCD. La caja de control se tienen cinco botones de derecha a izquierda como se muestra en la figura 2.2(b): el primer botón controla el contraste, los siguientes dos controlan el brillo y por último se tienen el botón que controla el



(a) Vista posterior.



(b) Vista frontal.

Figura 2.2 – Partes del Oculus Rift.

encendido y apagado del Oculus Rift. Al encenderse la caja de control se enciende un LED de color azul en la caja. La caja de control cuenta con entrada HDMI (entrada de vídeo), DVI (entrada de vídeo), USB mini-B (salida de datos del Oculus Rift) y entrada para fuente de poder DC, como se observa en la figura 2.2(b).

El Oculus Rift fue diseñado para tener un gran nivel de inmersión. Gracias al diseño del Oculus Rift es fácil bloquear la vista hacia el exterior en el mundo normal, dando como resultado una vista completa del mundo virtual. Emula la forma en que la vista funciona en el mundo real, agregando un gran campo de visión y mostrando distintos tipos de imagen en cada ojo. Por medio del sistema de sensores permite que dentro del mundo virtual se obtenga la vista requerida.

El sistema de sensores del Oculus Rift crea una interfaz natural intuitiva debido a que si el usuario desea ver una parte de un mundo virtual en específico solo necesitará girar la cabeza hacia esa dirección deseada y lo verá. Este sistema de sensores detecta la aceleración espacial en tres ejes y la velocidad de rotación en tres ejes, ambos agregan seis grados de libertad. Obteniendo los valores del sistema de sensores se puede representar cada gráfica del mundo virtual de tal forma que se puede obtener el valor actual del sensor o predecir su valor a partir del valor actual para asegurar que se despliegue de manera rápida los modelos tridimensionales en el mundo virtual con la perspectiva del usuario.

Normalmente una cámara de teléfono tiene 90 grados de campo de vista, Oculus Rift provee de un campo de vista mayor a los 90 grados de manera vertical y más de 110 grados de manera horizontal. Puede variar según la configuración establecida. Debido a los lentes especiales que utiliza Oculus Rift se puede obtener un mayor campo de visión a los 110 grados.

Los lentes que se utilizan en el Oculus Rift están diseñados de tal forma que ayuda a distorsionar la luz para evitar problemas enfoque, lo que ocasiona que el

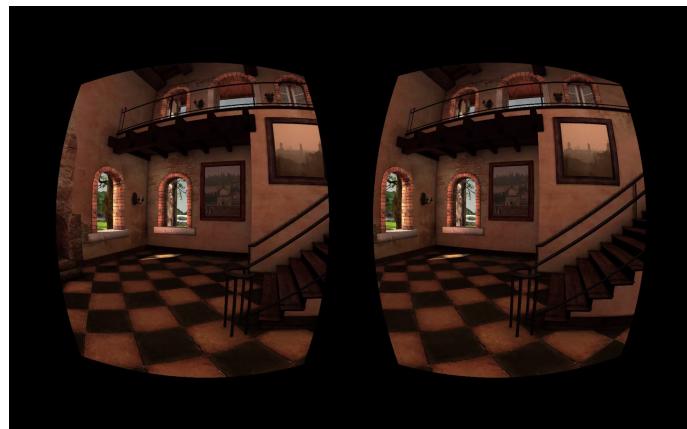
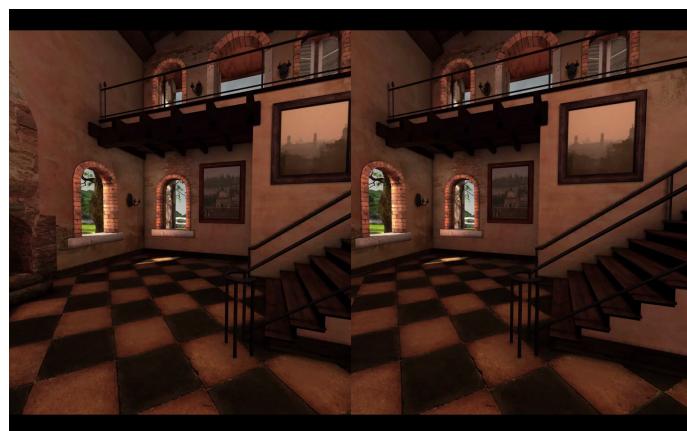
(a) Imagen estereográfica con distorsión *pincushion*.(b) Imagen estereográfica sin distorsión *pincushion*.

Figura 2.3 – Ejemplo de imagen con y sin distorsión de *pincushion*, tomado de la aplicación “Tuscany”, la cual es incluido al SDK que ofrece Oculus Rift.

panel se vea más amplio y ayuda a aumentar el campo de visión. Los lentes utilizados en Oculus Rift distorsionan la imagen de la pantalla LCD por medio de un efecto de lente de pescado. La manera en la que funciona es la siguiente: se envía una imagen distorsionada a la pantalla LCD, luego los lentes cancelan la distorsión dando como resultado una imagen clara y expandida. En la figura 2.3 se observa como se ve una imagen sin distorsionar en la pantalla LCD del Oculus Rift. El tipo de distorsión observada se llama *pincushion*¹⁹.

¹⁹Explicación sobre pincushion <http://toothwalker.org/optics/distortion.html>.

CAPÍTULO 3

LITERATURA EXISTENTE

En este capítulo se realiza un resumen de la investigación de los trabajos relevantes y que están relacionados con este trabajo. Se comentan los criterios de búsqueda utilizados para encontrar dichos trabajos. Después se realiza un estudio comparativo de cada una de las herramientas que fueron utilizadas en cada uno de los trabajos. Luego se realizan conclusiones sobre las tecnologías de cada uno de los trabajos existentes y al finalizar se hace un análisis sobre el área de oportunidad encontrado a partir del estudio comparativo realizado.

3.1 TRABAJOS EXISTENTES

Se realizó una exhaustiva investigación de trabajos de investigación que de alguna forma estén relacionados al trabajo presente. Dichos trabajos fueron encontrados gracias a *Google Scholar*¹, *IEEE*², *Springer*³, *ACM*⁴ y *ScienceDirect*⁵ usando las palabras clave *virtual reality* (por su traducción al inglés, realidad virtual), *training* (por su traducción al inglés, entrenamiento) y *learning* (por su traducción al inglés, aprendizaje).

¹Sitio de Google Scholar <http://scholar.google.com>.

²Sitio de IEEE <http://ieeexplore.ieee.org/Xplore/home.jsp>.

³Sitio de Springer <http://link.springer.com>.

⁴Sitio de ACM <http://dl.acm.org>.

⁵Sitio de ScienceDirect <http://www.sciencedirect.com>.

Entre los artículos encontrados, se encuentra un trabajo realizado por San Diego et al. [13] que describe un sistema de realidad virtual para entrenar a dentistas en donde resalta el uso de una tecnología sensorial que imita sensaciones del tacto. En dicho estudio trabajan con dos grupos de estudiantes de licenciatura para odontología en un laboratorio en donde prueban la realidad virtual usando: lentes para ver las imágenes 3D, cámara digital, dispositivo haptico (simula sensaciones) y un proyector para proyectar imágenes en 3D.

Al término de un periodo de que el grupo de estudiantes estuvieran practicando dentro de la plataforma de realidad virtual se les realizaron evaluaciones siguiendo ciertas reglas que son comúnmente realizadas para evaluar enseñanzas por medio de plataformas tecnológicas. Las evaluaciones dieron como resultado que los dos grupos de estudiantes que habían comenzado a utilizar la plataforma de realidad virtual tenían los mismos conocimientos que los que habían practicado con molduras dentales.

Chau et al. [4] discuten el ambiente 3D como potencial plataforma de aprendizaje tipo videojuego que pueda cambiar la perspectiva de los alumnos, lo cual los motiva al aprendizaje. En este artículo se habla sobre una evaluación empírica de aprendizaje en mundos virtuales utilizando una plataforma de videojuegos llamada SecondLife en donde ellos rentaron un espacio virtual para realizar dentro del juego pruebas de evaluación para ver como se comportaban los usuarios que exploran el área virtual contra los que ven un vídeo de la exploración virtual. En el hacen análisis con 105 personas concluyendo que las personas dentro del mundo virtual tienen mayor retención de aprendizaje.

Xu et al. [20] crearon un sistema de realidad virtual para simular evacuaciones de emergencias en casos de incendio. Su mayor aporte fue el desarrollo del modelo de flama y humo en el mundo virtual basado en el parámetro dado en la simulación de

fuego. El sistema virtual realiza una inmersión del usuario dentro del mundo virtual en el que se muestran métodos de evacuación para realizar una evacuación segura, esto con el fin de crear evacuaciones seguras y efectivas.

Lok et al. [7] hicieron una propuesta de un sistema de realidad virtual; en él estudiaron la manipulación de objetos para asistir el aprendizaje, entrenamiento y realización de tareas costosas o de mayor riesgo. Este estudio se llevó a cabo comparando las tareas realizadas (tal como manipulación de objetos) en el mundo virtual con las tareas realizadas en el mundo real. En sus resultados no encontraron gran diferencia significativa en la inmersión, la fidelidad de la visión usando el sistema de realidad virtual o diferencias entre los objetos reales y los virtuales.

Wyk y de Villiers [19] proponen una solución a la seguridad en la industria minera basada en un sistema de realidad virtual. Esto se debe al alto índice de mortalidad los trabajadores en la industria minera de dicha región, esto a causa de un mal o insuficiente entrenamiento. Wyk y de Villiers [19] resaltan que realizar entrenamientos de práctica fuera de el ambiente de trabajo solo limita el aprendizaje para controlar situaciones en los que puede llegar a salvar la vida de una o varias personas. Por medio de realidad virtual eliminaron los riesgos asociados en un ambiente real, sin eliminar la esencia de una escena real de riesgo.

Stansfield et al. [16] presentan un trabajo sobre el diseño y la implementación de una sistema de realidad virtual para apoyar el entrenamiento de múltiples usuarios para realizar tareas complejas en el cual se necesitan tomar decisiones críticas que son necesarias para cumplir con el objetivo; su enfoque fue en realizar un entrenamiento personal médico para respuestas de emergencia. El sistema de realidad virtual que desarrollaron tiene como característica la inmersión total de los usuarios que son identificados por cuerpo completo; en él se pueden hacer manipulaciones de movimiento de objetos virtuales, permitiendo a los usuarios actuar de forma natural

en el ambiente virtual. Dentro del sistema de realidad virtual se simula un ambiente inteligente en el que cualquier acción puede llegar a tener una consecuencia que ocurriría en algún ambiente real.

3.2 ESTUDIO COMPARATIVO

En esta sección se hará referencia a las características del desarrollo de software y la herramientas de hardware utilizados en cada uno de los trabajos relacionados. En el cuadro 3.1 y en el cuadro 3.2 se muestran los trabajos relacionados para cada característica que se desea comparar y se muestra si cumplen o no con los criterios establecidos.

San Diego et al. [13] realizaron un trabajo en el que involucraba la creación de un mundo virtual en donde se tenía que manipular una dentadura por medio de un sensor de tacto, luego este mismo es evaluado y se determina si se cumplió con el objetivo establecido. La forma en que se mostraban estas imágenes era por medio de una proyección para visualizar los objetos en 3D. Este trabajo para su propósito cumple el objetivo de un sistema de realidad virtual, sin embargo si se desea tener una aplicación más compleja, este solamente será muy limitado por lo que se ve en la proyección y no por el movimiento corporal del usuario.

Chau et al. [4] rentaron un espacio de un mundo virtual llamado SecondLife en él realizaron distintas actividades para medir el impacto del aprendizaje. El uso de este mundo virtual era por medio del uso del teclado y ratón. En este trabajo las evaluaciones que realizaron demostraron que el usuario tiene mayor retención de aprendizaje, aún sin tener un sistema de realidad virtual como tal implementado.

Xu et al. [20] desarrollaron un mundo virtual para evacuaciones en caso de incendios, esto para ayudar a crear evacuaciones seguras y efectivas. Las evacuaciones son evaluadas por su sistema de realidad virtual, sin embargo su sistema es muy limitado ya que el medio de interacción es por medio de un ratón.

Lok et al. [7] implementaron un mundo virtual para realizar una manipulación de objetos por medio de unos guantes con sensores, en donde se le pedía al usuario ordenar los objetos acorde a un color o tamaño. La inmersión fue por medio de unos lentes de realidad virtual. Los resultados de las evaluaciones de las actividades y de usabilidad demostraron que son similares a pruebas que realizaron en el mundo real. En este trabajo solamente mostraron actividades en las que el usuario manipula objetos por medio de guantes con sensores atribuyendo un nivel de inmersión agradable.

Wyk y de Villiers [19] discuten sobre una solución a la seguridad de la industria minera en África, en donde propone la implementación de un sistema de realidad virtual contando como características la evaluación de las acciones de usuarios dando como resultado consecuencias que pueden perjudicar al usuario en el mundo virtual. Sin embargo, estos solo pueden ser proseguidos utilizando un ratón y sin la posibilidad de tener un campo de visión dentro del mundo virtual, dando como resultado un nivel de inmersión no agradable para el usuario.

Stansfield et al. [16] presentaron una propuesta de un sistema de realidad virtual para entrenar personal médico. El sistema soporta múltiples usuario al mismo tiempo. Cada usuario puede utilizar el teclado y ratón para interactuar con el sistema de realidad virtual. El sistema de realidad virtual se compone de un grupo de sensores que determinan el movimiento del usuario. El sistema de realidad virtual hace evaluaciones de las actividades desempeñadas por el usuario. La inmersión con el sistema de realidad virtual se realiza por medio de unos lentes de realidad virtual. La interacción con los objetos se realiza por medio de un simulador de tacto que mide el agarre que realiza el usuario con un dispositivo. Este trabajo es uno de los más completos que se encontró en la investigación de trabajos existentes. Sin embargo la cantidad de sensores utilizados puede ser disminuida por medio de visión

Cuadro 3.1 – Características de desarrollo en software.

Trabajos	Control por dispositivo	Control por medio cuerpo	Implementación del mundo virtual	Evaluación de actividades	Evaluación de usabilidad
San Diego et al. [13]	Sensor de tacto	✗	✓	✓	✓
Chau et al. [4]	Teclado y ratón	✗	✗	✓	✓
Xu et al. [20]	Ratón	✗	✓	✓	✗
Lok et al. [7]	Guantes con sensores	✗	✓	✓	✓
Wyk y de Villiers [19]	Ratón	✗	✓	✓	✓
Stansfield et al. [16]	Teclado y ratón	✓	✓	✓	✓

Cuadro 3.2 – Características de herramientas en hardware.

Trabajos	Lentes de realidad virtual	Lentes con filtro para 3D	Cámara RGB	Simulador de tacto	Dispositivo de emisión de sonido	Proyección de objetos en 3D	Uso de pantallas
San Diego et al. [13]	✗	✗	✓	✓	✓	✓	✗
Chau et al. [4]	✗	✗	✗	✗	✓	✗	✓
Xu et al. [20]	✓	✗	✗	✗	✗	✗	✗
Lok et al. [7]	✓	✗	✗	✗	✗	✗	✗
Wyk y de Villiers [19]	✓	✗	✗	✗	✓	✗	✗
Stansfield et al. [16]	✓	✗	✗	✓ ⁶	✓	✗	✗

computacional o cámara de profundidad.

3.3 ÁREA DE OPORTUNIDAD

La mayoría de los trabajos presentados no se enfocaban en una interacción natural para el usuario, debido a que los dispositivos utilizados fueron controladores como teclados, ratones e incluso crearon dispositivos para determinar la presión que se tiene al hacer un agarre. La interacción con el mundo virtual debería ser de tal forma que el usuario olvida que existen esos dispositivos que son controladores para llevar a cabo actividades dentro del mundo virtual. Lo más correcto es que el usuario no debería interactuar conscientemente con los dispositivos.

CAPÍTULO 4

SOLUCIÓN PROPUESTA

En este capítulo se realiza el desarrollo del sistema de realidad virtual. En la sección de la metodología de desarrollo de software se discute la metodología a aplicar en el desarrollo de la solución propuesta. Después se explica en la sección de fases de desarrollo cada una de las fases por la que se compone la solución propuesta de este trabajo. Las fases discutidas en la sección de fases de desarrollo son las siguientes: exploración de tecnologías, planeación de implementación, diseño de prototipo y implementación. En la sección de exploración de tecnologías se realiza un análisis de las tecnologías que hay en el área de software y hardware que hay actualmente en el mercado. Después en la sección de planeación de implementación se narra la historia del usuario que utilizará el sistema de realidad virtual, esto con fines para después establecer un diseño de las etapas de desarrollo para llevar acabo la implementación del sistema de realidad virtual.

Luego en la sección de diseño de prototipo se discute el flujo general del sistema de realidad virtual, la máquina de estados del comportamiento del mundo virtual, las secuencias del mundo virtual y el flujo de la aplicación que obtiene la información del usuario, esto apartir de diagramas que se encuentran en el apéndice B. Además se describe el diseño del entorno 3D.

Por último, en la sección de implementación se llevan a cabo las etapas esta-

blecidas en la sección de planeación de implementación.

4.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE

La metodología de desarrollo de software seleccionada fue programación extrema. Beck y Andres [2] discuten la definición e implementación de la metodología de desarrollo de software llamada programación extrema. El enfoque de la *programación extrema* es realizar pequeñas entregas seguidas tras una iteración de pruebas con el fin de cumplir con el objetivo esperado de la entrega.

La programación extrema establece que dentro de la planeación se tienen que crear historias de usuario sobre como va a utilizar el software, ayudando a tener una mejor visión del alcance del software. Después se establece un análisis de la historia para establecer los pasos en los que se dividirá el desarrollo de las entregas del software. En cada una de estas entregas se realiza una iteración posterior, en donde dentro de esa iteración de realizan pruebas hasta establecer que ese mismo desarrollo establece los criterios esperados para ser entrega. En esta metodología establece que se tienen que tener juntas cada período de tiempo, en este caso se realizarán con mi asesor durante un período establecido. El desarrollo de dichas etapas se verán reflejados en un control de versiones llamado Git¹, que pueden ser consultados por medio de la herramienta en línea llamada GitHub².

Debido a las características de la metodología presentada, se eligió esta metodología de desarrollo de software para el desarrollo de la solución propuesta.

¹Sitio en el control de versiones Git <http://git-scm.com/>.

²Sitio en el que explican la herramienta GitHub <https://developer.github.com/guides/>.

4.2 FASES DE DESARROLLO

Las fases de desarrollo por las que se compone son de las siguientes:

Exploración de tecnologías

Se hace análisis de las tecnologías que actualmente están en el mercado. En él se discuten posibles alternativas que cumplen con el perfil deseado del prototipo y se realiza una revisión de cada una de las tecnologías viendo sus beneficios por medio de un estudio comparativo. Además se realiza una selección de las tecnologías presentadas a partir de criterios que cumplen con el perfil del trabajo que se espera realizar.

Planeación de implementación

Se desarrolla la historia de cómo el usuario se desenvolverá utilizando el sistema de realidad virtual. Con esta información se plantean las etapas por las que será desarrollado el sistema de realidad virtual que ocurre en la fase de implementación.

Diseño de prototipo

Se hace una revisión de las historias anteriormente realizadas. Esto ayuda a detectar los datos, métodos y propiedades con las que el sistema de realidad virtual necesitará. Se realiza un diagrama del flujo general de la información para tener una idea conceptual del flujo de los datos de todo el sistema de realidad virtual. Después se crean los diagramas de estados máquina incorporando los datos recopilados para determinar la transición de un estado a otro. Luego se realiza un diagrama de secuencia para identificar el orden de cómo será el procedimiento del flujo del sistema de realidad virtual. Por consiguiente se realiza un bosquejo del entorno 3D en el que se desarrollarán las actividades requeridas dentro del sistema de realidad virtual. Por último se realiza una

diagrama de flujo del procedimiento de la obtención de los movimientos del usuario.

Implementación

Se realiza el desarrollo de las etapas propuestas en la fase de planeación, tomando como referencia el diagrama de secuencias y diagramas de máquina de estados. Dentro de cada etapa se realizan pruebas para verificar que funcionen tal como fueron propuestos, además de ser requisito para pasar a la siguiente etapa.

Estas son las fases por la se compone el proyecto, sin embargo esto no quiere decir que esto es todo el alcance del proyecto. En los próximos capítulos se realiza el diseño y evaluación de análisis de tiempos de respuesta y usabilidad.

4.3 EXPLORACIÓN DE TECNOLOGÍAS

Primeramente, hay que establecer los puntos clave por los cuales serán buscadas las tecnologías. Sherman y Craig [14] mencionan que para tener una experiencia de realidad virtual se tienen cuatro elementos clave: el mundo virtual, la inmersión, la retroalimentación sensorial y la interactividad.

También mencionan que el mundo virtual es un espacio en el que se interactúa con objetos y con un conjunto de reglas establecidas; la inmersión es una sensación de estar dentro del ambiente virtual estimulando los sentidos del usuario por medio del mundo virtual; la retroalimentación sensorial provee información del movimiento y posición del usuario; la interactividad sucede cuando el mundo virtual reacciona a las acciones del usuario.

En el cuadro 4.1 se muestran las claves por las que se compone la realidad

Cuadro 4.1 – Claves de realidad virtual y tecnologías a explorar.

Claves de realidad virtual	Tipo	Necesidad
Mundo virtual	Software	Creación de entornos 3D
Inmersión	Hardware	Visualización del mundo virtual
Retroalimentación sensorial	Hardware	Obtener información del movimiento y posición del usuario
Interactividad	Software	Creación de interacción con el usuario en el entorno 3D

virtual según Sherman y Craig [14]. Además se incluyen los criterios de tipo de herramienta a buscar (ya sea hardware o software) y necesidad de lo que se busca que realice la herramienta.

4.3.1 MUNDO VIRTUAL

Por medio de Google se realizó una búsqueda sobre herramientas para la creación de modelos 3D, además de realizar un análisis sobre si realmente dichas herramientas se mantienen actualizados a los cambios de la tecnología, usando como referencia libros, foros de comunidades y documentación. Las herramientas deberán tener al menos las siguientes características: animación de los modelos 3D, edición de modelos 3D (cambiar tamaño, color), análisis de rendimiento del modelado, importación de activos y soporte a múltiples sistemas operativos.

Las herramientas encontradas fueron Unity³, Unreal Engine⁴ y Blender⁵, tal como se pueden observar en el cuadro 4.2. Unreal Engine no tiene soporte de uso en otros sistemas operativos que no sean Windows. Esto es importante debido a que se desea que el trabajo pueda ser descargado por cualquier persona y pueda ser usado ya sea para desarrollo o para utilizarlo. Mientras que Blender es un software de uso libre con un enfoque mayor al desarrollo de modelos 3D, sin embargo no cuenta con el soporte necesario para desarrollar un sistema de realidad virtual . Unity por un lado tiene como desventaja su licencia de pago, pero existe la opción de tener la licencia

³Sitio oficial de Unity <https://www.unity3d.com>.

⁴Sitio oficial de Unreal Engine <https://www.unrealengine.com/>.

⁵Sitio oficial de Blender <http://www.blender.org/>.

Cuadro 4.2 – Herramientas de desarrollo de entornos en 3D.

Nombre	Animación de modelos 3D	Editor de modelos 3D	Construcción de escenas	Herramientas para medir rendimiento	Importación de activos	Licencia de paga	Sistemas operativos	Nota
Unity	✓	✓	✓	✓	✓	✓ ⁶	Mac OS X y Windows	Tiene una grande y activa comunidad de desarrolladores.
Unreal Engine	✓	✓	✓	✓	✓	✓	Windows	El costo de la licencia es al exportar el proyecto a una aplicación.
Blender	✓	✓	✓	✗	✓	✗	Windows, Mac OS X y Linux	Tiene foros de discusión activos.

de no paga básica la cual es suficiente para el desarrollo de modelos 3D. Asimismo Unity cuenta con soporte para la creación de sistemas de realidad virtual gracias a las librerías que proporcionan terceros que desarrollan hardware de realidad virtual. La herramienta de desarrollo de entornos en 3D que es seleccionada es Unity, debido a su fácil uso de interfaz gráfica, su número de librerías de uso gratuito y por la facilidad de encontrar material para aprender a utilizarlo.

El día 19 de marzo del 2014 Unreal Engine anunció soporte para el sistema operativo Mac OS X, y esto ocurrió después de la selección de herramientas de desarrollo de entornos en 3D para el presente trabajo.

4.3.2 INMERSIÓN

Actualmente existen muchos dispositivos que ayudan crear la inmersión en la realidad virtual pero pocos tienen gran campo de visión que ayude a tener realmente una sensación de inmersión. Además pocos tienen incluido un sensor de movimiento de la cabeza y pocos cumplen con una latencia aceptable de entrega de información. Por medio de una búsqueda exhaustiva se encontraron las siguientes herramientas que se pueden observar en el cuadro 4.3. En el cuadro se pueden observar los criterios

de búsqueda utilizados para comparar las características de una tecnología con otra. Los dispositivos que se encuentran en el cuadro son los que actualmente se encuentran en el mercado.

En el cuadro 4.3 se observa que el Oculus Rift tiene como característica un mayor campo de visión comparado con las otras tecnologías, esto nos permite aumentar el nivel de inmersión dentro de un mundo virtual. Sin embargo tiene poca resolución, aunque para fines del presente trabajo es suficiente. La compañía que desarrolló Oculus Rift también provee un SDK para el desarrollo de aplicaciones de realidad virtual utilizando la herramienta de Unity. Además el precio del Oculus Rift es más accesible que las otras tecnologías por las que es comparado. El Oculus Rift es la herramienta de inmersión seleccionada debido a las características que tiene.

4.3.3 RETROALIMENTACIÓN SENSORIAL

Ahora se discuten las tecnologías que ayudarán a realizar la detección del movimiento y posición corporal, para ello se distinguen tres características que ayuden a detectarlo. La primera característica es la detección de profundidad. Gracias a esta característica se puede determinar si la persona está moviendo hacia adelante y atrás, además de ayudar a obtener un modelo 3D del usuario y las posiciones exactas de las articulaciones y partes del cuerpo⁷ del usuario. Es cierto que es posible determinar la posición de una persona por medio de una cámara convencional con algoritmos de visión computacional y detectar las articulaciones y partes del cuerpo del usuario pero debido a que no es el objetivo a corto plazo de este proyecto se optó por buscar una tecnología existente que cubra esta necesidad y que requiera menos poder de cómputo. El seguimiento de posición de dedos de la mano es una característica muy importante si realmente se quiere interactuar con objetos dentro

⁷Las articulaciones y partes del cuerpo que son de interés en esta solución propuesta son las rodillas y el torso del usuario.

Cuadro 4.3 – Dispositivos para ver entornos virtuales.

Nombre	Precio	Resolución de pantalla	Campo de visión horizontal	Tamaño de pantalla	Peso	Sensor de movimiento de cabeza	Audio	Tipo de entrada	Batería	Nota
Oculus Rift ¹	300 USD	640 × 800	más de 90 grados	7 pulgadas	379 gramos	✓	✗	DVI ² /HDMI ³ y USB ⁴	✗	Tiene una pantalla para cada ojo. También provee un SDK para integrar las lentes en Unity3D y Unreal Engine.
Carl Zeiss Cinemizer ⁵	789 USD	870 × 500	30 grados	0.39 pulgadas	120 gramos	✓	✓	HDMI y USB	✓	Provee un SDK para obtener datos del seguimiento de la cabeza.
Sony HMZ-T3W ⁶	1299 EUR	1280 × 720	45 grados	Desconocido	320 gramos	✗	✓	HDMI y USB	✓	Se puede usar de manera inalámbrica. Tiene una pantalla para cada ojo.
Silicon Micro Display ⁷	799 USD	1920 × 1080	39 grados	0.74 pulgadas	106 gramos	✗	✓	HDMI y micro USB	✗	Se puede agregar una batería externa.

¹<http://www.oculusvr.com/rift>²<http://www.es.wikipedia.org/wiki/Dvi>³http://es.wikipedia.org/wiki/High-Definition_Multimedia_Interface⁴<http://es.cinemizer.zeiss.com>⁵<http://www.sony.es/electronics/head-mounted-display/hmz-t3w>⁶<http://www.siliconmicrodisplay.com/st10801.html>

Cuadro 4.4 – Características de dispositivo por cámara de profundidad, seguimiento de articulaciones y partes del cuerpo y dedos.

Dispositivo	Profundidad	Articulaciones	Dedos
Kinect	✓	✓	✗
Guantes con sensores de flexibilidad	✗	✗	✓
Leap Motion ¹	✗	✗	✓

¹<https://www.leapmotion.com/product>

del mundo virtual, lo cual agregaría experiencia de inmersión más rica en el mundo virtual.

Como se puede observar en el cuadro 4.4 las tecnologías presentadas tienen enfoques distintos, por ejemplo el Kinect puede detectar las partes del cuerpo y profundidad gracias a su proyector y sensor de luz infrarroja, los guantes con sensores de flexibilidad son ajustables a la medida, ya que es factible construir desde cero un guante con sensores de flexibilidad que pueden ser conseguidos en alguna tienda de electrónica, además de ajustar los umbrales para determinar eventos. Leap Motion tiene como característica principal la capacidad de detectar el movimiento de las manos y dedos de manera independiente utilizando bajo costo computacional y dando una baja latencia, aunque tiene sus altibajos: en algunas posiciones de las manos puede detectar perfectamente la posición de los dedos y en otras se confunde detectando como un dedo múltiples dedos.

El Kinect es la herramienta de retroalimentación sensorial elegida, debido a que posee una cámara de profundidad lo cual puede ahorrar el tiempo de análisis computacional para determinar la posición exacta del usuario. Además existe una comunidad que desarrolla una librería de código abierto que ofrece funcionalidades para detectar las posiciones exactas de las partes del cuerpo.

4.3.4 INTERACTIVIDAD

Debido a que anteriormente se seleccionaron las tecnologías Kinect y Oculus Rift, se tomarán en consideración la herramienta de Processing y Unity. Unity posee un característica llamada *scripting* el cual es el ingrediente principal del mundo virtual, esto se debe a que por medio de los scripts se pueden definir los comportamientos de los modelos 3D creados con Unity. Este es el porque se tomó en consideración como herramienta de interactividad.

Por otro lado Processing posee una librería llamada SimpleOpenNi el cual ofrece la posibilidad de obtener la información del Kinect por medio de funciones prefabricadas, además de que también existe la librería UDP el cual permite crear conexiones hacia otras aplicaciones a través del protocolo de transporte UDP. Esto tiene relación ya que por medio de la obtención de los datos del Kinect la aplicación en la que se despliega el mundo virtual recibirá la información de esos datos por medio de una conexión por UDP; es por eso que fue seleccionada también esta herramienta como herramienta de interactividad.

4.4 PLANEACIÓN DE IMPLEMENTACIÓN

La historia de usuario se realiza en base a el objetivo general y objetivos específicos los cuales pueden verse en el capítulo de introducción. La historia comienza con una persona llamado Miguel el cual hace el papel del usuario del sistema realidad virtual que se pretende realizar. Dicha historia se encuentra en el apéndice A que se encuentra en la página 91.

Se puede observar en la figura 4.1 el flujo de la información dentro del sistema de realidad virtual de una manera general. En él se puede observar que primero se

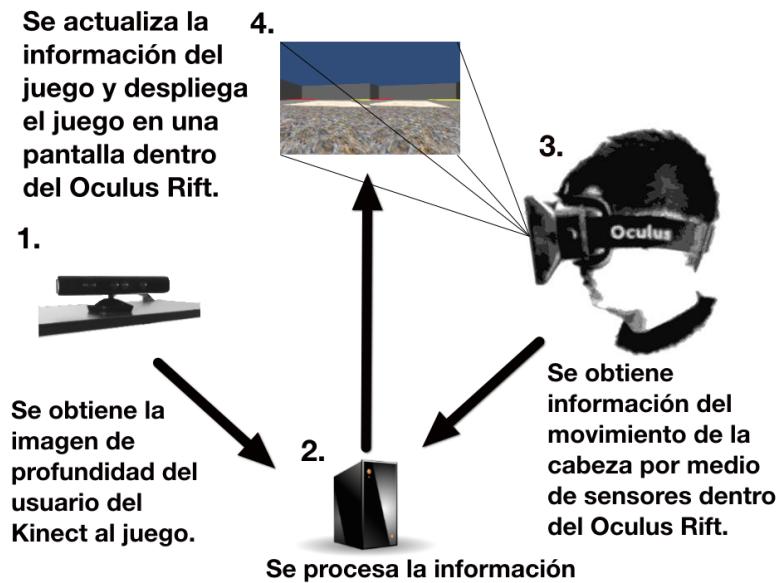


Figura 4.1 – Diagrama de flujo general del sistema de realidad virtual.

obtiene la información de la persona por medio del Kinect. Luego dentro de una computadora se procesa la información de la imagen de profundidad del Kinect para determinar qué acciones realizó el usuario. Por otro lado, Oculus Rift manda la información de su sistema de sensores que se encuentra en el dispositivo hacia la computadora en donde se está ejecutando el sistema de realidad virtual. Por último se actualizan los datos dentro del mundo virtual y se muestra el mundo virtual en el Oculus Rift.

Las etapas que se identificaron para el plan de trabajo fueron determinadas a partir de la historia del usuario utilizando el sistema de realidad virtual. Las etapas son las siguientes:

1. Implementación de mundo virtual de prueba con Oculus Rift.
2. Establecimiento de comunicación entre aplicación que comunica el Kinect y mundo virtual.
3. Extracción de datos del Kinect.

4. Desarrollo de mundo virtual del sistema de realidad virtual.
5. Desarrollo de la evaluación de acciones del usuario.
6. Establecimiento del formato por el cual serán enviados los datos del usuario a la aplicación del mundo virtual.
7. Desarrollo de lógica del mundo virtual.

El orden de presentación de estas etapas es por el cual se van a llevar cabo y tienen base a la importancia de los objetivos que fueron identificados en este trabajo.

4.5 DISEÑO DE PROTOTIPO

A base de la redacción la redacción de la historia del usuario utilizando el sistema de realidad virtual se realizó un análisis de los datos, métodos y propiedades de los datos que se identificaron. Los datos identificados de la aplicación del mundo virtual fueron los siguientes:

Cliente UDP Establece una conexión entre la aplicación del Kinect y de la aplicación del mundo virtual.

Dirección IP Establece la dirección por el cual se establecerá la comunicación por el protocolo UDP.

Puerto Establece el puerto por el cual se establecerá la comunicación por el protocolo UDP.

Mensaje recibido Se encarga de recibir el mensaje recibido dentro de la aplicación del mundo virtual, mismo que contendrá información sobre el movimiento y posición del usuario.

Camina Se encarga de ser un verificador de que si el usuario está caminando o no.

Orientación Se encarga de contener la dirección hacia la que se dirige el usuario.

Detección de usuario Se encarga de contener la información que informa si el usuario es detectado.

Levanta mano Se encarga de informar si el usuario levanta la mano derecha.

Colores Se encarga de contener los colores a visitar rojo, amarillo, verde y azul.

Colores visitados Contiene un conteo de los colores visitados.

Tiempo inicial Contiene la información del tiempo inicial de la actividad.

Nivel activo Contiene el nivel en el que se encuentra.

Mensaje de instrucciones Contiene las instrucciones de la actividad.

Mensaje de puntuajes Contiene el mensaje que se mostrará en conjunto con el tiempo obtenido.

Los métodos identificados dentro de las redacción de usuario fueron los siguientes:

Cambiar de nivel Se encarga de cambiar de una escena a otra. Por ejemplo, de cambiar una escena donde se espera que el usuario levante la mano derecha al usuario a una escena donde se comienza la actividad.

Salir del sistema Se encarga de salir de la aplicación del mundo virtual.

Obtener información si usuario camina Se encarga de entregar la información si el usuario camina.

Asignar información si usuario camina Se encarga de asignar la información recibida si el usuario camina.

Obtener información de orientación de usuario Se encarga de entregar la información de la orientación del usuario.

Asignar información de orientación de usuario Se encarga de asignar la información de la orientación del usuario que es recibida.

Obtener información si el usuario es detectado Se encarga de entregar la información si el usuario es detectado.

Asignar información si el usuario es detectado Se encarga de asignar la información recibida si el usuario es detectado.

Obtener información si el usuario levanta la mano Se encarga de entregar la información si el usuario levanta la mano derecha.

Asignar información si el usuario levanta la mano Se encarga de asignar la información si el usuario levanta la mano derecha.

Obtener colores Se encarga de entregar la información de los colores.

Obtener colores visitados Se encarga de entregar los colores visitados.

Ordenar colores aleatoriamente Se encarga de ordenar los colores de manera aleatoria.

Verifica color Se encarga de verificar el color por el que el usuario ha visitado, en caso de no ser el color requerido moverá al usuario a la posición inicial. Cuando visita el usuario el color correcto este método cambia la cantidad de lugares visitados.

Mover usuario a posicion inicial Se encarga de mover al usuario dentro del mundo virtual a la posición inicial.

Cuadro 4.5 – Datos identificados de la aplicación del mundo virtual.

Clase de dato	Nombre de variable
ClientUDP ¹	ClienteUDP
String ²	Dirección IP
	Puerto
	Mensaje recibido
	Nivel activo
	Mensaje de instrucciones
	Mensaje de puntuajes
	Orientación
Boolean ³	Camina
	Detección de usuario
	Levanta mano
List ⁴	Colores
Int ⁵	Colores visitados
Float ⁶	Tiempo inicial

¹ClientUDP es una clase del cliente UDP.²String es una clase de cadena de caracteres.³Boolean es una clase para expresar valores *booleanos*: “verdad” y “falso”.⁴List es una clase para un listado de elementos.⁵Int es una clase para representar números enteros.⁶Float es una clase de un número decimal en representación de punto flotante.

Cambiar instrucciones Se encarga de cambiar las instrucciones por contador de lugares visitados y viceversa, esto al momento que el usuario levante la mano derecha.

Obtener mensaje colores visitados Realiza el formato del mensaje de los colores visitados y el contador de colores visitados.

Verifica posición de usuario Verifica que el usuario se encuentre dentro del área requerida.

Actualiza datos de usuario Se encarga de actualizar los datos del usuario, tal como el movimiento y su posición.

El siguiente paso es obtener las propiedades de los datos tomando la decisión a partir de la información por el que es descrito el dato. Las propiedades identificadas de la aplicación del mundo virtual se pueden observar en el cuadro 4.5.

Ahora se verán los datos identificados de la aplicación que procesa la información del usuario. Los datos identificados fueron los siguientes:

Cliente UDP Para establecer una conexión con la aplicación por la cual se muestra el mundo virtual.

Dirección IP Para establecer la dirección IP por el cual se establecerá la comunicación por el protocolo UDP.

Puerto Para establecer el puerto por el cual se establecerá la comunicación por el protocolo UDP.

Kinect Por medio del cual se obtiene la información directamente del dispositivo de Kinect.

Mensaje de “camina” Es el formato del mensaje que contiene la información si el usuario camina.

Mensaje de “orientación” Es el formato del mensaje que contiene la información de la orientación del usuario.

Mensaje de “usuario detectado” Es el formato del mensaje que contiene la información si el usuario es detectado.

Mensaje de “usuario levanta mano” Es el formato del mensaje que contiene la información si el usuario levanta la mano.

Mensaje enviado Es el mensaje que contiene todos los mensajes en un formato CSV⁸.

Rodilla derecha Contiene las coordenadas de la rodilla derecha del usuario detectado.

Rodilla izquierda Contiene las coordenadas de la rodilla izquierda del usuario detectado.

⁸Variables separadas por comas en texto plano.

Cuello Contiene las coordenadas del cuello del usuario detectado.

Mano derecha Contiene las coordenadas de la mano derecha del usuario detectado.

Orientación Contiene la orientación por la cual se encuentra el usuario.

En lo siguiente se identifican los métodos identificados de la aplicación que procesa la información del usuario.

Obtención orientación del torso Se encarga de obtener la información de la orientación del torso del usuario, el cual entrega información de la dirección hacia donde va el usuario.

Detección caminata Se encarga verificar si se detectó que el usuario hizo un movimiento de caminar por medio de las posiciones de las rodillas derecha e izquierda.

Nuevo usuario Se encarga de avisar cuando nuevo usuario es detectado y después de ser llamado se procede a extraer la información del usuario.

Usuario perdido Se encarga de avisar cuando el usuario detectado es perdido de vista.

Usuario visible Se encarga de avisar que el usuario sigue siendo visible y activo.

Las propiedades que se identificaron dentro de la aplicación que procesa la información del usuario fueron determinadas en base a la función de cada uno de los datos. Los datos identificados de la aplicación que procesa la información del usuario se pueden observar en el cuadro 4.6.

En la figura B.1 que se encuentra en el apéndice B en la página 94, se puede observar el diagrama máquina de estados del sistema de realidad virtual, en cada

Cuadro 4.6 – Datos identificados de la aplicación que procesa la información del usuario.

Clase de dato	Nombre de variable
ClientUDP	ClienteUDP
String	Dirección IP
	Puerto
	Mensaje de “camina”
	Mensaje de “orientación”
	Mensaje de “usuario detectado”
	Mensaje de “usuario levanta mano”
	Mensaje enviado
Kinect ¹	Kinect
PVector ²	Rodilla derecha
	Cuello
	Mano derecha
PMatrix3D ³	Orientación

¹Kinect es una clase de Kinect por medio del que se obtienen datos del Kinect.

²PVector es una clase es para almacenar vectores y su uso común es para almacenar coordenadas.

³PMatrix3D es una clase de tipo matriz el cual sirve para almacenar arreglos de 4×4.

transición se tiene una variable el cual su nombre es representativo y expresa las variables involucradas para pasar de un estado a otro. El primer estado es “Comienza”, en él dirá se espera que el usuario presione el botón para comenzar, después el usuario se encontrará en el estado “EsperaUsuario” en donde se espera a que el usuario sea detectado y que levante la mano derecha para continuar al siguiente estado. Luego el usuario entra al estado “Actividades” en el que hará una actividad que involucra visitar áreas de colores según el orden en el que son presentados, mismos que se representan dentro de el área en la que se encuentra el usuario. Al visitar todos los colores el usuario pasará al estado de “Resultados”; en él se presentará el tiempo que tardo en realizar las actividades el usuario dentro del estado “Actividades”, además verá un texto en él que si desea reiniciar la actividad, el usuario podrá hacerlo levantando la mano derecha.

En las figuras B.2 y B.3 que se encuentran en el apéndice B en las páginas 95 y 96, se muestra el diagrama se secuencias de la lógica del mundo virtual, en donde se puede ver en detalle el proceso de las transiciones de un estado a otro del mundo virtual. En dicho diagrama se tienen seis instancias que son representadas como clases: *Comienza*, *Estado mundo virtual*, *Usuario*, *Espera usuario*, *Actividades*

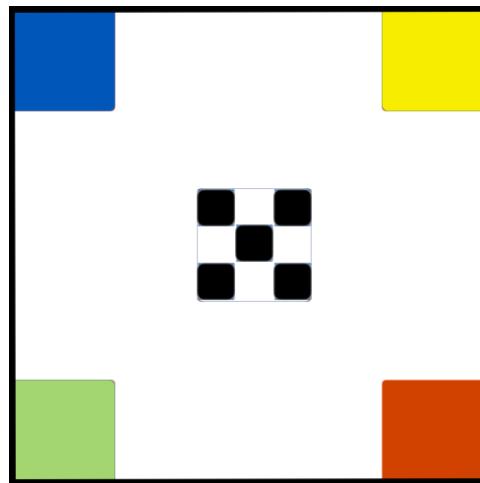


Figura 4.2 – Vista áerea del entorno virtual.

y *Resultados*.

El diseño del mundo virtual en el estado de “Actividades” se puede ver en la figura 4.2. Se encuentra desde una perspectiva áerea de las cuatro áreas de colores que el usuario irá a visitar al comenzar el estado de “Actividades”, tal como son redactadas dentro de la historia y tomando como referencia el cuadro central como la posición de inicio del usuario. En el perímetro se pueden observar líneas negras las cuales representan los muros que ayudarán a que el usuario esté dentro de un área limitada del mundo virtual.

En la figura B.4 que se muestra en el apéndice B en la página 97, se puede observar un diagrama de flujo de la aplicación que obtiene la información del movimiento y posición del usuario. Al comenzar la aplicación este se encuentra dentro de un ciclo que itera un número de decisiones y procesos. La primer decisión es si el usuario es detectado, en caso de serlo se obtiene una imagen de profundidad del usuario, misma que es capturada con el Kinect. Seguidamente se establecen las variables por omisión, estas variables son las que se almacenarán la información como la orientación del usuario, si el usuario camina, si el usuario levanta mano derecha y si el usuario es detectado. Después se verifica si es posible detectar la orientación del

usuario, en caso de detectarla se establece la orientación del usuario en una variable. Lo mismo sucede con las siguientes dos decisiones: se verifica si el usuario camina y si el usuario levanta la mano, en caso de serlo se les asigna el valor correspondiente. Luego se realiza un proceso de enviar en un formato CSV las variables. Por consiguiente se vuelve a iterar el mismo flujo. La utilización de diagramas de flujo es la más natural para representar este tipo de solución debido a la naturaleza en la que se desarrolla en la herramienta de Processing.

4.6 IMPLEMENTACIÓN

La implementación del mundo virtual de prueba con Oculus Rift consiste en el desarrollo de un entorno dentro de Unity como se observa en la figura 4.3, se tuvieron que ver algunos conceptos [3] para entender como está compuesta la interfaz y cómo comenzar a desarrollar en ella. Se comenzó con la implementación de un plano en 3D con una textura de tierra, solamente fue crear un objeto llamado *Game Object* después ajustando la vista de la cámara de desarrollo, lo que ayudó a determinar las dimensiones ideales para el plano las cuales pueden ser configuradas gracias a una opción llamada *Inspector* el cual es una propiedad del Game Object, una vez establecido el plano se agregó una esfera para que pueda ser como punto de referencia al momento de hacer una prueba con las lentes Oculus Rift, posteriormente se realizó el mismo procedimiento de configuración hecha con el plano. Después dentro de los foros de desarrolladores de Oculus Rift se encontraron manuales creados por Antonov et al. [1] y Yao et al. [21], mismos que sirvieron como guías para implementar una escena dentro de Unity utilizando Oculus Rift.

Al ejecutar la escena creada en Unity, se verifica que el seguimiento del movimiento de la cabeza se reflejarán dentro del mundo virtual.

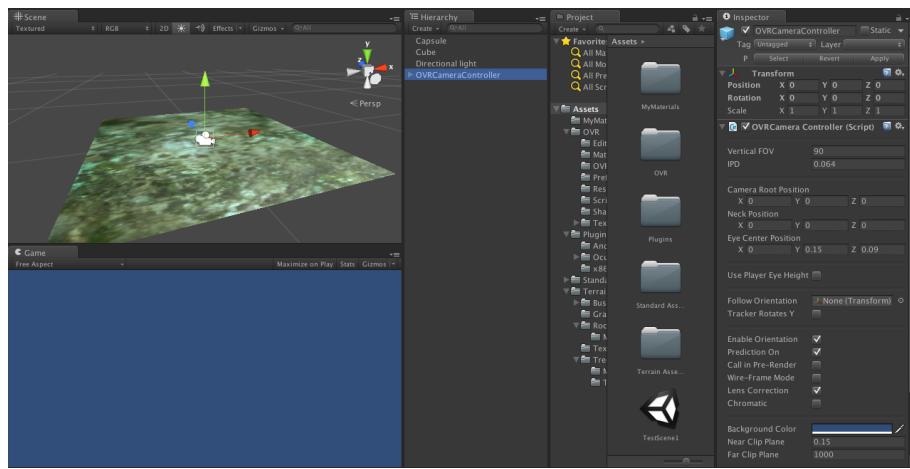


Figura 4.3 – Herramienta de Unity en el que se manipulan los modelos 3D.

El establecimiento de comunicación entre aplicación que comunica el Kinect y mundo virtual se compone por dos programas. Uno de los programas es creado en Processing, el objetivo de éste es implementar un cliente UDP que enviará la información de prueba a otro programa en Unity. El programa en Unity que recibirá la información del programa en Processing es otro de los programas que se implementan. En Unity los programas son llamados *scripts* dentro de este script se implementa un cliente UDP que estará escuchando la información que se envía desde el programa en Processing. En el capítulo de antecedentes se discute sobre el protocolo de transporte UDP y librerías utilizadas en Processing para establecer la comunicación.

En la figura 4.4 se observa una escena creada en Unity en donde se tienen dos campos de texto, uno para establecer la dirección IP y el otro para establecer el puerto. Ambos datos sirven para establecer la dirección IP y el puerto al cual el cliente UDP del script creado estará escuchando los datos que reciba del programa en Processing. Al recibir el dato este sobreescribirá el texto “No message received yet” que significa no se ha recibido ningún mensaje aún. Para comenzar a escuchar por información al ejecutar la escena en Unity se presiona al botón “Start listening”

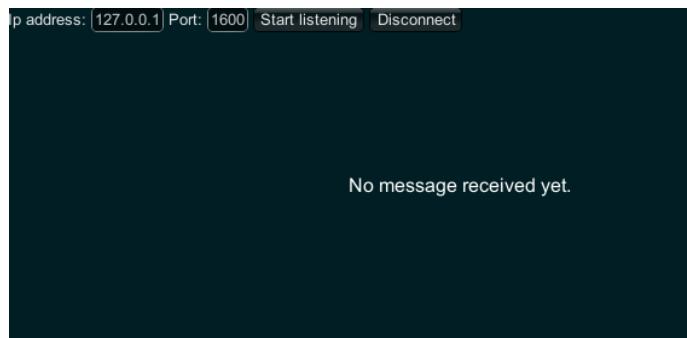


Figura 4.4 – Escena de prueba de comunicación con cliente UDP en Unity.

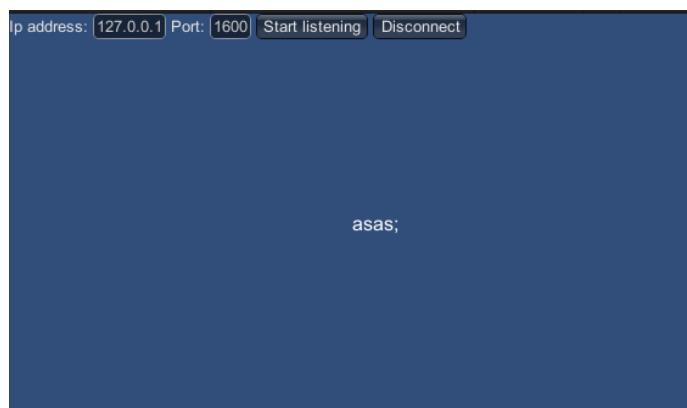


Figura 4.5 – Prueba de comunicación con cliente UDP en Unity.

el cual significa comienza a escuchar, en caso de querer para el programa se presiona el botón “Disconnect” para dejar de escuchar por datos.

En la figura 4.5 se muestra que el cliente UDP recibió un mensaje de prueba y sustituye en texto anterior, mismo que es enviado desde el cliente UDP del programa en Processing.

La extracción de datos del Kinect se puede realizar por medio de la librería de SimpleOpenNi en Processing, mismo que se discute en el capítulo de antecedentes. El Kinect por medio de su cámara infrarroja ayuda a tomar imágenes de profundidad al utilizar el proyector de luz infrarroja sobre el usuario. En SimpleOpenNi existe un método llamado `enableDepthImage` el cual ayuda a obtener un mapa de profundidad de lo que se encuentra en la parte frontal del Kinect. La forma en la que se representa



(a) Imagen capturada desde la cámara de profundidad del Kinect.
(b) Imagen capturada desde la cámara RGB del Kinect.

Figura 4.6 – Ejemplo de imagen de profundidad tomada desde el Kinect.

el mapa de profundidad es por medio de un vector de enteros en el que cada entero tiene como valor la distancia que alcanza a proyectar la cámara infrarroja dentro de un rango de 500 milímetros a 8000 milímetros. El tamaño del vector obtenido dentro de una imagen de profundidad es equivalente a la cantidad de píxeles de la imagen capturada del Kinect que puede ser desde una resolución de 640×480 píxeles (307200 píxeles) a 1280×1024 píxeles (1310720 píxeles) según la resolución de la imagen será más exacto aunque por otro lado será más lento la velocidad de procesamiento por la cantidad de píxeles. En la figura 4.6 se puede observar una imagen de profundidad de un torso y brazos tomada desde el Kinect.

Desarrollo del mundo virtual del sistema de realidad virtual consiste en la creación de modelos 3D y ajustarlos según el diseño anteriormente realizado. Para ello se creó un objeto del tipo terreno, este será la base en la que se desenvolverá el mundo virtual. En la figura 4.7 se puede observar el objeto del tipo terreno. En él se le asigna las dimensiones adecuadas, en este caso se colocarán 100 metros de ancho por 100 metros de largo.

Para evitar que el usuario salga del área existen varias técnicas pero la más común es usando paredes, en este caso de crearon cuatro objetos con base a un cubo para que éste luego se le dé una forma de pared cambiando sus dimensiones.

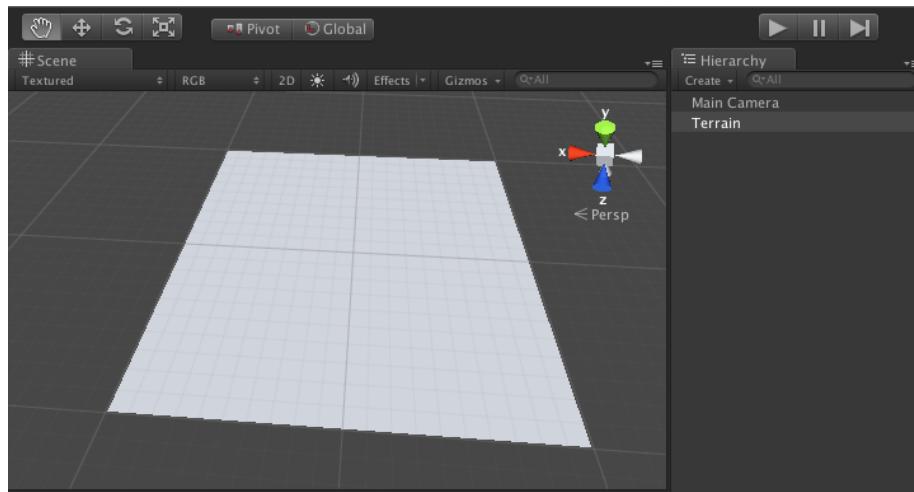


Figura 4.7 – Objeto tipo terreno en Unity.

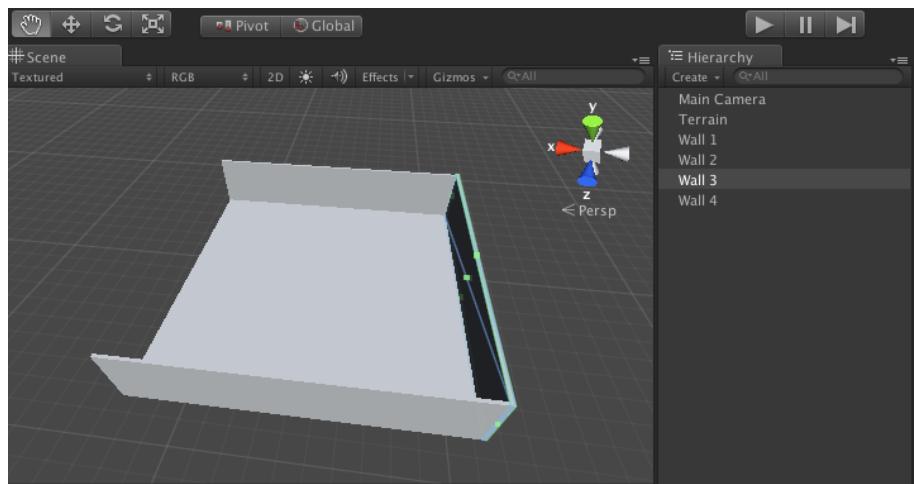


Figura 4.8 – Implementación y manipulación de paredes para limitar al usuario.

En la figura 4.8 se puede observar la implementación y manipulación de las paredes colocadas en el mundo virtual.

En la historia de usuario se había propuesto que se visitarán áreas de colores, para ello se crearon materiales personalizados para asignar el color a los objetos que representarán las áreas. Para representar el área se crearon cuatro objetos tipo planos los cuales, como su nombre lo dice, son planos representados en 3D. Cada plano se le asignó un color. En la figura 4.9 se puede observar la implementación y asignación de colores de cada plano.

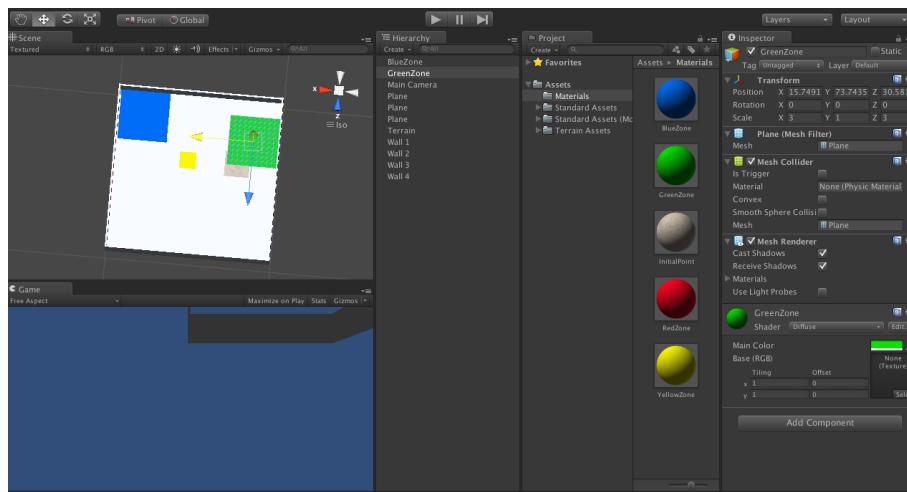


Figura 4.9 – Implementación de los planos y colores correspondientes.

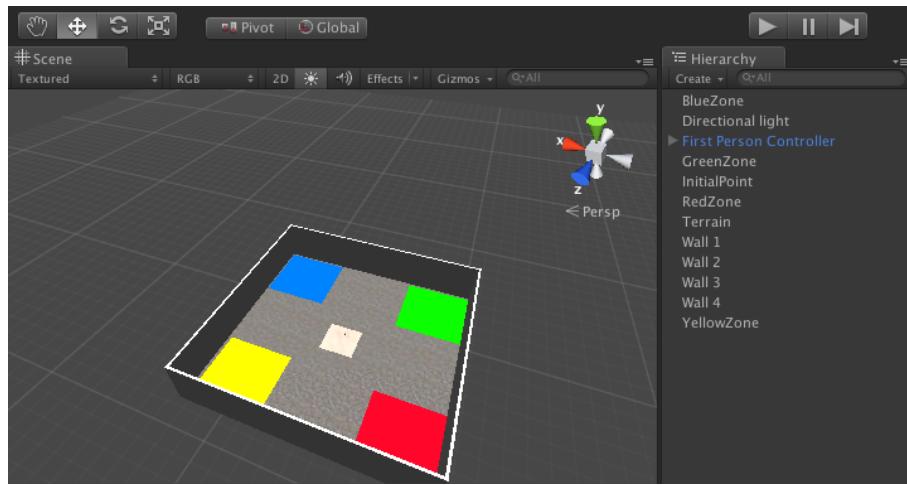


Figura 4.10 – Modelo del mundo virtual terminado.

Para representar el punto de inicio del usuario se creó un quinto plano que es colocado en medio de los cuatro planos de colores creados. En la figura 4.10 se encuentra el modelo del mundo virtual, tal como fue propuesto en el diseño del mundo virtual.

El desarrollo de evaluación de acciones del usuario implica la obtención de la imagen de profundidad de usuario. La librería de SimpleOpenNi cuenta con un método llamado `enableUser`, esto habilita el seguimiento de las partes del cuerpo del usuario que es captado dentro de la imagen de profundidad captada por el Kinect.



(a) Posición que facilita la detección de las partes del cuerpo.
 (b) Posición dificulta la detección de las partes del cuerpo.

Figura 4.11 – Posiciones fáciles y difíciles para la detección de las partes del cuerpo.

Además SimpleOpenNi detecta la cantidad de usuarios en el campo de visión del Kinect y los etiqueta por un número de identificación que pueden ser accedidos dentro de una lista. Al ser detectado un nuevo usuario se llama un método llamado `OnNewUser`; éste lo que hace es llamar a otro método llamado `isTrackingSkeleton` que comienza a realizar un seguimiento del nuevo usuario detectado tomando como parámetro el número de etiqueta del nuevo usuario detectado.

Por medio de la posición que se observa en la figura 4.11 se facilita al algoritmo que determina los puntos de las partes del cuerpo, mismo que está incorporado en la librería SimpleOpenNi. En la figura también se pueden observar puntos rojos mismos que representan cada una de las partes del cuerpo identificados.

Después de ser detectado el usuario e identificado las partes del cuerpo de manera correcta, por medio de un valor de confidencia en cual la librería SimpleOpenNi proporciona para identificar la confidencia de la información recibida del algoritmo que obtiene e identifica las partes del cuerpo. Este lo que hará es llamar el método `obtenOrientacionUsuario`, tomando como parámetro la etiqueta por la cual el usuario es identificado. Después se obtiene la posición del torso del usuario en el sistema de coordenadas representado en la imagen de profundidad.

dad captada del usuario. Una vez obtenida dicha posición se llama a un método `getJointOrientationSkeleton` tomando como parámetro la etiqueta del usuario, una constante que representa el torso del usuario y una matriz de cuatro por cuatro (4.1) en el cual se almacena la traslación y rotación del usuario:

$$R = \begin{bmatrix} r_{00} & r_{10} & r_{20} & r_{30} \\ r_{01} & r_{11} & r_{21} & r_{31} \\ r_{02} & r_{12} & r_{22} & r_{32} \\ r_{03} & r_{13} & r_{23} & r_{33} \end{bmatrix}. \quad (4.1)$$

Luego para obtener la orientación del usuario, se utilizó la formula de ángulos de Euler (4.2) en el eje y en la matriz obtenida:

$$\theta_y = (-r_{02}, \sqrt{r_{21}^2 + r_{22}^2}). \quad (4.2)$$

Como se muestra en la figura 4.12(b), al momento de estar el usuario en 90 grados en el eje y con respecto a su sistema de coordenadas, éste estará en una posición neutral. Con neutral significa que el usuario no está haciendo ningún movimiento hacia la derecha o izquierda. También se tuvo que establecer un rango de grados para determinar si el usuario sigue en la posición neutral, esto por medio de un experimento empírico. Los grados en y en su sistema de coordenadas tienen que ser mayores a 70 grados y menores a 110 grados para decir que el usuario se encuentra en una posición neutral.

En la figura 4.12(a) se muestra que al momento que el usuario está a 110 grados en el eje y con respecto a su sistema de coordenadas, éste estará en una orientación hacia la izquierda. Se realizó el mismo procedimiento empírico para determinar el rango de valor para decir que el usuario se encuentra en una orientación izquierda. Para determinar que el usuario se encuentra en una orientación izquierda éste tiene

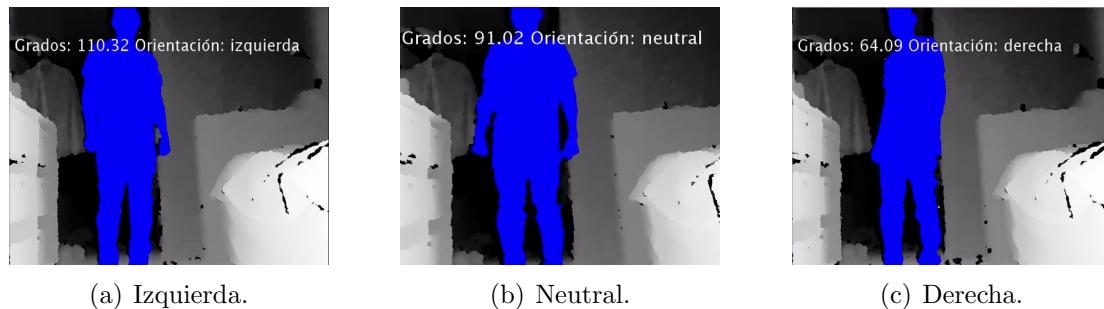


Figura 4.12 – Detección de la orientación de usuario en las posiciones izquierda, neutral y derecha.

que tener más de 110 grados en y en su sistema de coordenadas.

La figura 4.12(c) se muestra el usuario está a 64 grados en el eje y con respecto a su sistema de coordenadas. Para determinar que el usuario está en la orientación derecha los grados en su eje y tienen que ser menores a 70 grados.

Después de obtener la orientación del usuario se llama al método que identificará si el usuario camina el cual es llamado `obtenerSiUsuarioCamina`. Dentro del método se obtienen las posiciones de la rodilla izquierda y derecha en el sistema de coordenadas. Dentro de este método se evalúa si el usuario realiza un paso y apartir de eso se dice que el usuario camina, para ello se tiene que cumplir al menos una de las siguientes reglas.

1. La posición de la rodilla derecha en el eje y del sistema de coordenadas más el umbral tienen que ser menor que la posición de la rodilla izquierda.
2. La posición de la rodilla izquierda en el eje y del sistema de coordenadas más el umbral tienen que ser menor que la posición de la rodilla derecha.

Nótese que se usa un umbral dentro de estas reglas. Esto sirve para que en caso de existir una diferencia entre las posiciones de las rodillas, éstas deberán estar arriba de un cierto valor establecido para ese umbral y así determinar que el usuario realiza un paso. En la figura 4.13 se puede observar cuando el usuario es identificado

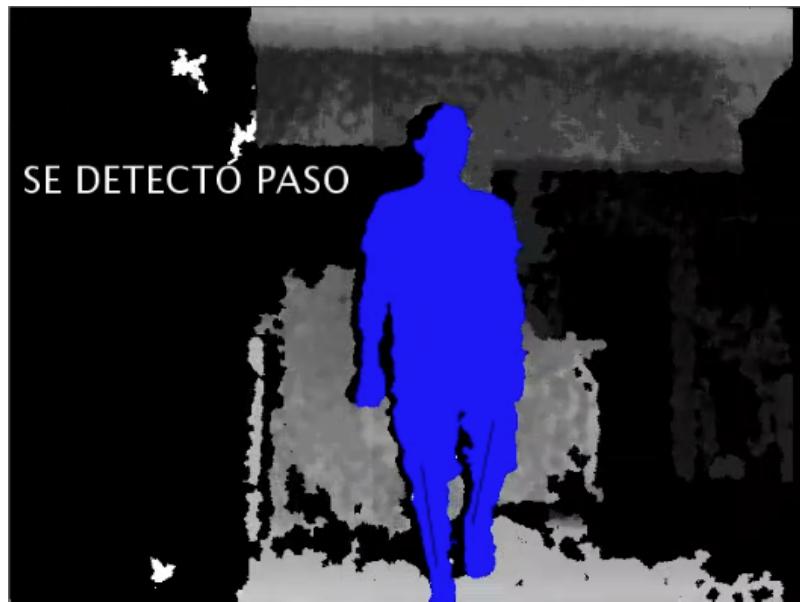


Figura 4.13 – Se detecta el usuario caminando.

que camina por medio de las reglas establecidas.

Por último, se determina si el usuario levanta la mano derecha por medio de un método llamado `deteccionManoDerecha`. Esto se hace tomando como posición de referencia el cuello por medio de la librería de SimpleOpenNi, luego se obtiene la posición de la mano derecha de la misma forma que la posición del cuello. Después tomando el valor del eje *y* del cuello si es éste mayor al valor del eje *y* de la posición de la mano derecha éste dirá que el usuario levanta la mano derecha.

El formato que se lleva acabo para enviar la información del usuario a través del programa en Kinect es CSV. Cada valor separado por comas tiene su propio significado. El primer valor es si el usuario es detectado enviando un *uno* cuando el usuario es detectado, o *cero* cuando el usuario no es detectado. El segundo valor es la orientación del usuario se recibe como valor *neutral*, *derecha* o *izquierda* dependiendo de la orientación del usuario. El tercer valor es si el usuario camina, en él se recibe un *uno* cuando el usuario es detectado que camina o *cero* en caso contrario. Por último, el cuarto valor es si el usuario levanta la mano derecha, en él

se recibe un *uno* en caso de que el usuario haya levantado la mano derecha o *cero* en caso contrario.

El desarrollo de la lógica del mundo virtual se constituye de cuatro estados como se habían mencionado en la sección planeación de implementación. Para cada estado se creó una clase. Además se crearon la clase “Usuario” el cual se encarga de obtener la información del usuario y la clase “EstadoMundoVirtual” para administrar los estados del mundo virtual. En Unity cada clase y estado serán considerados como scripts, los scripts son los que contienen la lógica del mundo virtual. Las escenas son objetos 3D, objetos vacíos o interfaces gráficas, en cada escena se puede adjuntar un script que ayuda a crear interacción entre el usuario y el mundo virtual.

Estado comienza Como se muestra en la figura 4.14(a) “Comienza” se implementa un botón que contiene como texto “Comienza”. Dentro del script se tiene un método que es llamado al ser el botón presionado, este método cambia el estado del mundo virtual a través del script llamado “EstadoMundoVirtual”. Éste crea una instancia de la clase “EstadoMundoVirtual”, si éste existe debido a que el usuario reinició la actividad, éste no se crea de nuevo. Dentro de este script “EstadoMundoVirtual” se llama al estado “EsperaUsuario” para comenzar a hacer el procedimiento de detección de usuario. Gracias a la clase “Usuario” se obtiene la información del movimiento y posición del usuario, en el script de “EstadoMundoVirtual” se crea un objeto de esta clase que provee los métodos para obtener la información del usuario que pueden ser accedidos por cualquiera de los estados. Dentro de la clase “Usuario” se tiene un cliente UDP que obtiene y actualiza los datos que envía la aplicación que obtiene los datos del usuario por el Kinect.

Estado espera usuario En la figura 4.14(b) se puede observar el texto “Detec-

tando usuario” en este se espera que el usuario sea detectado. Una vez siendo detectado el usuario se cambiará el texto del la escena a “Levanta tu mano derecha para comenzar” como se muestra en la figura 4.14(c), en él se espera que el usuario levante la mano derecha para comenzar a realizar las actividades dentro del mundo virtual.

Estado actividades Al momento que el usuario levanta la mano derecha se comienza el siguiente estado “Actividades”, en él, el usuario se encuentra en una escena de un cuarto con cuatro áreas de color rojo, amarillo, azul y verde. Al entrar al estado “Actividades” se comienza a guardar el tiempo inicial en el que el usuario está comenzando la actividad el cual nos sirve para determinar en el estado “Resultados” el tiempo de desempeño del usuao. El tiempo inicial esta almacenado en el script “EstadoMundoVirtual”. Frente al usuario se muestran instrucciones como las que se observan en la figura 4.14(d). Las instrucciones muestran el orden en el que se tienen que visitar las áreas de colores. En él, el usuario tendrá que realizar movimientos similares al de subir escaleras para caminar dentro del mundo virtual. Para moverse hacia los lados el usuario tendrá que mover sus hombros hacia la derecha o izquierda, esto también funciona al mover el cuello hacia una dirección. Si el usuario levanta la mano derecha, las instrucciones en el mundo virtual son removidas y son reemplazadas por el mensaje “Faltan 4 colores por visitar.” tal como se muestra en la figura 4.14(e). El número de los colores visitados cambia a como se van visitando las áreas de colores correctas. Visitar un área de un color incorrecto tomando como referencia la secuencia de colores establecida ocasionará que el usuario sea llevado a la posición inicial (el área central). En caso de que el usuario visite todas las áreas de color de manera correcta este se pasará al estado “Resultados”.

Estado resultado En el estado “Resultado” se mostrarán los resultado del tiempo

que tardó en realizar visitar las áreas de color el usuario. En la figura 4.14(f) se muestra un ejemplo del estado “Resultado”. Para obtener el tiempo de desempeño se hizo una sustracción del tiempo inicial con el tiempo en el que fue accedido a este estado.



Figura 4.14 – Estados del mundo virtual del sistema de realidad virtual.

CAPÍTULO 5

EXPERIMENTOS

En este capítulo se discuten el diseño y evaluación de los experimentos propuestos que tratan sobre la usabilidad y latencia del sistema de realidad virtual. Esto con el fin analizar posibles problemas del sistema de realidad virtual con usuarios. Asimismo proponiendo alguna solución a los problemas que se presenten.

5.1 DISEÑO

En esta sección se discuten los experimentos que serán evaluados en la sección de evaluación. Los experimentos son propuestos tratan sobre usabilidad y latencia del sistema de realidad virtual. El primer experimento trata sobre la utilización del protocolo pensamiento en voz alta para identificar problemas de usabilidad en el sistema de realidad virtual, además se realiza un plan de gestión de tiempo de la aplicación del mismo. El segundo experimento es sobre analizar la latencia de la comunicación que existe entre el envío y recepción de datos dentro del sistema de realidad virtual, se proponen las métricas y se establece un límite de latencia máxima. El tercer experimento es una evaluación heurística del sistema de realidad virtual, el enfoque en esta evaluación es sobre la experiencia del usuario dentro del sistema de realidad virtual. Un ejemplo de la experiencia de usuario sería si el dispositivo Oculus Rift responde de manera adecuada en el mundo virtual al girar el cuello,

sin retardos. Por último se propone una plantilla de evaluación para identificar los problemas de usabilidad.

5.1.1 PROTOCOLO DE PENSAMIENTO EN VOZ ALTA

El protocolo de pensamiento de voz en alta [10] es un método que nos permite obtener información sobre la usabilidad de una aplicación. Dicho método propone que el usuario tiene que realizar un conjunto de tareas y al realizar las tareas el usuario tiene que decir lo que piensa, hace, siente y mira. La actividad que se obtiene del usuario puede ser escrita o grabada por audio o vídeo. La única regla es no orientar al usuario y permitirle que use la aplicación de tal forma que permita evaluar si el usuario es capaz de realizar correctamente una tarea establecida. Durante el tiempo en que el usuario realice las tareas deberá ser necesario obtener notas para determinar cuales pueden ser algunos de los problemas en la usabilidad de la aplicación, aunque por otro lado si es grabada es posible analizar detenidamente la actividad del usuario y realizar notas más completas.

En este experimento se realizarán pruebas utilizando el protocolo de pensamiento en voz en alta con tres usuarios. La sesión para cada prueba tendrá una duración aproximada de 30 minutos, en los que se divide la lectura de las instrucciones del uso del sistema de realidad virtual con una duración apróximada de 10 minutos, la colocación y ajuste del dispositivo Oculus Rift con una duración apróximada de 5 minutos y la utilización del sistema de realidad virtual con una duración apróximada de 15 minutos.

Cuando se inicia la sesión con el usuario se le comenta que su voz será grabada y en caso en que no esté de acuerdo, se tendrá que escribir la actividad del usuario.

Al usuario se le proporcionará el instructivo del uso del sistema de realidad

virtual para su lectura, dicho instructivo se presenta en el apéndice C de la página 98. Luego el usuario proseguirá a colocarse el dispositivo Oculus Rift y se le menciona que puede tener asistencia en caso de necesitar ajustar el dispositivo.

Después de colocarse y ajustarse el dispositivo Oculus Rift, el usuario utilizará el sistema de realidad virtual y se comenzará a grabar o escribir según la decisión del usuario. Si es grabada la voz del usuario, al terminar la sesión con el usuario se realizará un análisis de la grabación y se realizarán apuntes sobre cuales fueron los problemas que obtuvo el usuario. En caso de ser escrita la conversación, solamente se considerarán las notas importantes que fueron de mayor impacto al momento de realizar las tareas por parte del usuario. Al terminar el usuario se le agadecerá su participación dentro de la evaluación. Al analizar la actividad del usuario se determinarán los puntos por los que el usuario tuvo problemas de usabilidad y para cada punto identificado se explicará una conclusión sobre una solución a ello.

5.1.2 ANÁLISIS DE LATENCIA

El nivel de latencia se mide desde que se determinan las acciones del usuario hasta cuando las acciones del usuario hayan tomado efecto dentro del mundo virtual. Es importante medir la latencia del sistema de realidad virtual, ya que con ello se podrán identificar problemas que eviten que el usuario tenga un nivel de inmersión apropiado para experimentar la realidad virtual.

La medición de la latencia será de manera modular en cada programa que se compone el prototipo. El prototipo está compuesto por dos programas: el programa que extrae, evalúa y determina qué acción se ha realizado a partir los datos del sensor de retroalimentación y el programa que realiza la simulación y recibe los datos por parte del programa que extrae los datos del sensor de retroalimentación. Para obtener la latencia se necesitarán obtener los tiempos desde que se trajeron

los datos hasta la acción que tomó efecto dentro de la simulación; se creará un archivo para almacenar los tiempos de cada programa para que luego pueda ser procesado por un programa que evalúe las latencias.

5.1.3 EVALUACIÓN HEURÍSTICA

Una evaluación heurística es una inspección de la usabilidad de un sistema siguiendo un listado de reglas para identificar problemas de usabilidad dentro del diseño de interfaz de usuario. Los encargados de realizar la evaluación heurística son los evaluadores los cuales evalúan el listado de reglas que son llamadas heurísticas. Los métodos de evaluación heurística son rápidos y de bajo presupuesto debido a que utilizan un conjunto de reglas (heurísticas) simples y fáciles de identificar. En el trabajo propuesto por Nielsen y Molich [11] demuestra lo efectivo que es la evaluación heurística, la cual puede ser evaluada por cualquier persona.

Actualmente existen varios tipos de estándares de evaluación heurística para evaluar páginas web, instructivos, aplicaciones móviles, ambientes virtuales, realidad virtual, etcétera, los cuales están basados en las heurísticas propuestas por Nielsen y Molich [11]. Para este trabajo es adecuado realizar una evaluación heurística del ambiente virtual y realidad virtual.

Sutcliffe y Gault [17] discuten sobre una propuesta de un conjunto de heurísticas y un método de evaluación experta dicha propuesta es una evaluación heurística para realidad virtual, el cual es basado en el trabajo de Nielsen y Molich [11]. Esta evaluación heurística propuesta por Sutcliffe y Gault [17] hace un énfasis en la evaluación de ambientes virtuales. Además de intentar unificar una evaluación formativa con la evaluación de la presencia en la realidad virtual. En su trabajo Sutcliffe y Gault [17] realizan dos casos de estudio utilizando las heurísticas y el método de evaluación que propusieron.

Sutcliffe y Gault [17] explican en breve las heurísticas siguientes:

1. **Atado natural.** Interacción debe enfocarse en la expectación del usuario con la interacción en el mundo real lo más apegado posible. Idealmente el usuario debería estar inconsciente que la realidad es virtual. La interpretación de esta heurística dependerá sobre la naturaleza del requerimiento y la esencia de la presencia de usuario y atado a la realidad.
2. **Compatibilidad con las tareas y dominio del usuario.** El ambiente virtual y comportamiento de objetos deberá corresponder lo más cerca posible a la expectación del mundo real de los objetos al usuario; su comportamiento; y la comprensión intuitiva para realizar la tarea.
3. **Expresión natural de acción.** La representación de la presencia propia en el ambiente virtual debe permitir al usuario actuar y explorar de manera natural y sin restricción de acciones físicas normales. La calidad del diseño puede ser limitado por la disponibilidad de dispositivos. Si el sensor de retroalimentación está ausente, expresiones naturales van a sufrir inevitablemente.
4. **Estrecha coordinación de acción y representación.** La representación de la presencia propia y la manifestación del comportamiento en el ambiente virtual debe ser leal a las acciones del usuario. El tiempo de respuesta entre el movimiento del usuario y actualización de la pantalla del ambiente virtual deben ser inferior de 200 ms¹ para evitar los problemas de mareo.
5. **Retroalimentación realista.** El efecto de las acciones del usuario en los objetos del ambiente virtual deben ser inmediatamente visibles y conforme a las leyes de la física y las expectaciones perceptuales del usuario.

¹ms refiere a la abreviación de milisegundos.

6. **Puntos de vista leales.** La representación visual del mundo virtual deberá mapear la percepción normal del usuario, y el punto de vista deberá cambiar por el movimiento de la cabeza sin tener retrasos de renderización.
7. **Navegación y soporte de orientación.** El usuario deberá siempre ser capaz de buscar donde está en el ambiente virtual y volver a posiciones conocidas, preestablecidas. Acciones no naturales como volar a través de las superficies pueden ayudar pero éstas tienen que ser juzgados en un canje con naturalidad (ver las heurísticas 1 y 2).
8. **Entrada clara y puntos de salida.** El significado de la entrada y salida del mundo virtual deben ser comunicados claramente.
9. **Salidas consistentes.** Cuando el diseño comprometido es usado se deberán ser consistentes y claramente marcados, por ejemplo sustitución intermodal y las acciones de energía para la navegación.
10. **Soporte para aprender.** Objetos activos deberán tener pistas y si es necesario explicarse ellos mismos para promover el aprendizaje del ambiente virtual.
11. **Turnos claros.** Donde el sistema inicializa deberá ser claramente señalado y convenciones establecidas de los turno claros.
12. **Escencia de presencia.** La percepción del usuario estando en el mundo real deberá ser lo más natural posible.

El método de evaluación que proponen Sutcliffe y Gault [17] sigue las recomendaciones de Nielsen y Molich [11] con algunas modificaciones.

Para que los evaluadores realicen la evaluación heurística tienen que explorar la aplicación, para ello es necesario establecer un conjunto de tareas a realizar dentro del mundo virtual de los cuales luego ayudarán a encontrar problemas. Después por

medio del uso de las heurísticas vistas anteriormente se interpretarán y clasificarán los problemas. La evaluación heurística dependerá de las limitaciones que se tengan dentro del mundo virtual.

Adicionalmente Sutcliffe y Gault [17] propusieron un paso más para evaluar el sistema de realidad virtual y ambiente virtual, en donde se busca que las acciones realizadas en el mundo real por medio de la tecnología de realidad virtual sean representadas de manera correcta en el mundo virtual, dados los dispositivos utilizados. Este paso se lleva acabo cuando el evaluador ha realizado el listado de tareas y explora el ambiente virtual en donde toma notas de la presencia o ausencia de características en a las siguientes categorías a las que se asocie.

- Operación de la presencia de usuario. El usuario puede ser representado en el mundo virtual por un cursor o más comúnmente por una mano o un avatar completo. La presencia puede ser controlada por una variedad de dispositivos desde un ratón en 3D, una bola en el espacio, guantes de flexibilidad, trajes de inmersión de cuerpo completo, entre otros. La presencia del usuario y los controles dentro de la aplicación pueden causar muchos problemas siendo que se provee casi perfecta representación del usuario de las acciones en el mundo virtual. La idoneidad de la presencia necesita ser juzgada en relación con la acción natural del usuario. Para una simple navegación, la presencia puede ser no necesario; para manipulaciones, una mano virtual es usualmente necesario.
- Falta retroalimentación háptica. Verdaderos prototipos virtuales no tienen retroalimentación háptica (sensación de tocar), por lo que la presencia de usuario puede pasar a través de representaciones de objetos sólidos. Para eliminar la ausencia de la retroalimentación háptica, muchas aplicaciones usan la retroalimentación visual con algoritmos de detección de colisión para estimular a los usuarios cuando objetos son seleccionados o han sido seleccionados. Pro-

blemas causados por la ausencia de la retroalimentación háptica pueden ser observados con complejas manipulaciones y tareas físicas.

- Técnicas interactivas. Muchos ambientes virtuales implementan controles que permiten a los usuarios volar a través de los ambientes virtuales para buscar y seleccionar objetos distantes por una técnica llamada propagación de rayos (por su traducción al inglés como *ray-casting*). Esto puede ser tomado más allá proporcionando efectos mágicos como *snap-to* para que se puede saltar a objetos cercanos con la mano del usuario. Estos efectos pueden causar problemas de usabilidad cuando están probablemente diseñados.
- Gráficas realistas. Los ambientes virtuales pueden que no se puedan representar tal como fueron propuestos en el prototipo, desde que la mayoría de las aplicaciones no son representadas de manera fotorrealista a detalle. Sin embargo algunas evidencias sugieren que las personas pueden realizar las tareas naturalmente sin la necesidad de una representación de detalle visual. Detalle gráfico será importante para desplegar información y para tareas cuando el ambiente del sistema es visualmente complejo.

Una vez que el evaluador haya completado el conjunto de tareas, este podrá realizar la auditoría de la tecnología.

Estos problemas son interpretados con las heurísticas para evaluar la calidad de presencia y diagnosticar las características del diseño responsable por los problemas encontrados. La siguiente lista de verificación es una guía para la atribución de problemas de características de diseño:

- Visualización gráfica, profundidad 3D o perspectiva de distorsión, pobre resolución de imagen. Indicado por dificultad de percepción.

- Moviendo y manipulando la presencia del usuario, subdividido en el hardware que está siendo usado (guantes, ratón 3D, etc.) y la representación del usuario en el ambiente virtual. Indicado por la dificultad de navegación y manipulación.
- Interacción con herramientas y objetos en el ambiente virtual. Indicado por intentos de acciones no exitosas; o por pobre retroalimentación que confunde a los usuarios.
- Características ambientales. Parte del ambiente en el que fue creado hay efectos inesperados como mover a través de las paredes y objetos flotantes.
- Interacción con otros controles como menús flotantes.
- Problemas del hardware como una pantalla montada en la cabeza (HMD) y lentes con pantalla (*shutter glasses*).

Una vez que el evaluador haya diagnosticado los problemas asignándolos a las heurísticas y características de diseño, la etapa final es para clasificar la gravedad de los errores por heurística. Se dan indicaciones según la gravedad de los problemas identificados, clasificándolos desde un pobre diseño con un impacto severo, así como el resultado de una tarea fallida a un problema mínimo probablemente curable por medio de indicaciones al usuario. Esta clasificación refleja el número de errores asignado por cada heurística y es el juicio del evaluador quien corresponda a la gravedad de estos errores en una escala de cuatro puntos.

La escala de la gravedad de los errores es la siguiente:

- Grave. El problema encontrado hizo imposible completar la tarea de manera exitosa.
- Molesto. El problema va a interrumpir la tarea del usuario pero la mayoría de

los usuarios podrán a aprender como lidiar con el error dada una explicación, y algunos podrán aprender a lidiar con ello con el tiempo.

- Distraído. El problema va a interrumpir la tarea del usuario pero la mayoría de los usuarios podrán descubrir como arreglarlo relativamente rápido dada una indicación.
- Inconveniente. El problema puede interrumpir la tarea del usuario pero la mayoría de los usuarios encontrarán la solución sin ayuda.

5.2 RESULTADOS

En esta sección se discuten los resultados de los experimentos propuestos. Los experimentos que fueron propuestos son: protocolo de pensamiento en voz alta, análisis de latencia y evaluación heurística de usabilidad. En cada uno de los experimentos se da una explicación sobre las métricas utilizadas para medir los resultados; además se argumenta cada resultado obtenido. Las discusiones obtienen su base a través de una representación gráfica de los resultados obtenidos de los experimentos.

5.2.1 PROTOCOLO DE PENSAMIENTO EN VOZ ALTA

La evaluación del protocolo de pensamiento en voz alta se le aplicó a tres usuarios, todos los usuarios aceptaron ser grabados por audio. A continuación se verá un resumen de los puntos que se identificaron por medio del audio que fue grabado al momento que los usuarios utilizaron el sistema de realidad virtual.

Usuario 1 Al iniciar la evaluación el usuario tuvo buena impresión al usar el sistema de realidad virtual, en donde el usuario hablaba que el mundo virtual

era cercano a la realidad. Después de iniciar a realizar las actividades en el que se espera que el usuario levante la mano derecha para cambiar el mensaje de instrucciones a mostrar, el usuario se confundió sobre su uso. El usuario mantuvo levantada su mano derecha por un tiempo en el que la interfaz gráfica cambiaba los mensajes de instrucciones y mensaje de ayuda de colores faltantes a visitar; mientras que la manera correcta de realizarlo es levantando solamente una vez la mano derecha para cambiar de mensaje. Además el usuario comentó que al caminar un poco más rápido, el sistema no podía mantener el mismo ritmo al que él caminaba y eso ocasionaba la sensación de desesperación. Al girar los hombros el usuario comentó que eso ocasionaba que girará en el mundo virtual, además de argumentar de que estaba confundiendo el mundo virtual con la realidad.

Usuario 2 Despues de que el usuario inicializará el mundo virtual se confundió utilizando la interfaz gráfica por medio del levantamiento de la mano derecha, esto al cambiar el mensaje de instrucciones a mensaje de ayuda de colores faltantes a visitar y viceversa. Se le hizo difícil leer el orden de los colores dentro del mensaje de las instrucciones, esto debido a que en los lentes de Oculus Rift no los logró apreciar bien el usuario. Sintió desesperación debido a que el movimiento de sus piernas no correspondía a la velocidad en la vida real.

Usuario 3 El usuario no tuvo problemas al cambiar el mensaje de instrucciones al mensaje de ayuda de colores faltantes a visitar y viceversa, esto al levantar la mano derecha. Sin embargo tuvo problemas con seguir las instrucciones y tardó tiempo visitar las áreas correspondientes. Al realizar varios movimientos distintos de sus piernas descubrió unos defectos en el sistema de realidad virtual. Esto es al momento de mover sus rodillas en forma circular en su misma posición y el otro fue al mantener levantada una pierna.

Los problemas que encontraron los usuarios fueron sobre la velocidad de los pasos dentro del mundo virtual, detección de otros movimientos como pasos y problemas de uso de interfaz gráfica por medio de levantamiento de mano derecha.

La velocidad del movimiento de las piernas dentro del mundo virtual es debido a que se estableció una velocidad constante del movimiento del jugador al ser una vez detectado el usuario que camina en el mundo real.

La detección de otros movimientos como pasos se debe a que se realiza un patrón similar al movimiento de las rodillas al caminar y esto ocasiona a que sea detectado como si estuviera caminando. También sucede al intentar realizar un paso y permanecer con el pie de manera estática en el aire, esto se debe a que el movimiento es detectado como un paso y cada paso en el sistema de realidad virtual es detectado como si el usuario caminará.

Al momento de realizar esta prueba los usuarios tuvieron problemas al ser detectados por el Kinect de manera repentina, esto es debido a la interreferencia de rayos de luz infrarroja del sol² mismo que se estaba reflejando detrás del área en la que se estaba realizando el experimento.

El problema de la velocidad de los pasos dentro del mundo virtual se puede solucionar creando un evento que puede ser llamado cada vez que haya nueva información del usuario para actualizar su información dentro del mundo virtual, ya que actualmente se actualiza la información por período preestablecidos dentro de un script (parte del SDK de Oculus Rift), mismo que captura el movimiento de la cabeza del usuario.

El problema de la detección de los pasos se puede solucionar por medio de la detección independiente de los pasos de cada pie que se realicen. Por medio del

²Muestran experimento del uso del Kinect a la luz del sol. http://dasl.mem.drexel.edu/wiki/index.php/KINECT_in_direct_sunlight.

cambio de reglas de cada paso podría evitar confusiones con otros movimientos; una regla podría ser que al momento que el usuario levante el pie se encuentre una etiqueta que espere hasta que regrese el pie a donde estaba anteriormente.

El problema del uso de la infertaz gráfica por medio de levantamiento de la mano derecha podría ser por medio de una ilustración que se anime al momento que el usuario levante la mano derecha y que alrededor de la misma ilustración que haya una barra de progreso que al llenarse pueda permitir que el usuario cambie de mensajes a mostrar.

5.2.2 ANÁLISIS DE LATENCIA

La evaluación de análisis de latencia es importante debido a que de esta forma se pueden encontrar cuellos de botella de flujo de información dentro de la aplicación; un alto nivel de latencia puede hacer que el usuario tenga síntomas de mareo utilizando el prototipo. Esto se debe al retardo que ocasionaría entre las acciones que realizaría el usuario en el mundo real en comparación con el mundo virtual dando dichos síntomas de mareo, además de ocasionar una mala experiencia de usuario dentro del mundo virtual.

Sutcliffe y Gault [17] en su trabajo hablan sobre un límite de latencia que es tolerado para la percepción del ojo humano que existe entre el sensor de retroalimentación y el mundo virtual el cual es de 200 ms mientras que Sherman y Craig [14] habla que la latencia tolerada es de 100 ms. Sutcliffe y Gault [17] hablan sobre un límite de latencia tolerado para la percepción del ojo humano que existe entre el sensor de retroalimentación y el mundo virtual, sin embargo no muestran algún método para analizar la latencia y la evaluación realizada es a base el juicio de los evaluadores.

Sherman y Craig [14] discuten la importancia de tener una latencia baja debido a que el nivel de realismo es reducido cuando las entradas del usuario son lentas, ellos comentan que en la realidad virtual la lentitud se mide en ms en donde, 16 ms es una latencia muy buena en tiempo de respuesta mientras que 100 ms es la mayor respuesta aceptada. Considerando el trabajo de Sutcliffe y Gault [17] junto con el de Sherman y Craig [14], en esta evaluación se tomarán en consideración como latencia aceptable 100 ms, sin embargo se tomarán a consideración las opiniones de los usuarios durante las evaluaciones heurísticas, abriendo la posibilidad de que mayores latencias puedan ser aceptadas.

Este experimento consta de tres evaluaciones de latencias; la primera evaluación (fase A) es al hacer la determinación de la acción realizada por el usuario a partir de los datos obtenidos del sensor de retroalimentación; la segunda evaluación (fase B) es cuando la simulación del entrenamiento recibe una acción realizada por el usuario; la tercera evaluación (fase C) es al momento de que las acciones realizadas por el usuario están siendo representadas dentro del mundo virtual.

Se obtuvieron tres muestras de las latencias para realizar tres evaluaciones de latencias utilizando el sistema de realidad virtual del sistema de realidad virtual. En la figura 5.1 se puede observar una gráfica tipo pastel, en donde se representan los porcentajes de cada una de las fases por las que se divide el total de las latencias obtenidas. En el se puede observar que el menor porcentaje de latencia es de 0.22 % (fase A), el cual se tiene al obtener la información y evaluar las acciones del usuario determinada por el usuario. Mientras que el mayor porcentaje de latencia es de 54.50 % (fase B), el cual se tiene al recibir la acción dentro de la simulación de entrenamiento. Por último, el porcentaje intermedio de latencia es de 45.26 % (fase C), el cual se obtiene al momento que las acciones realizadas por el usuario sean representado dentro del mundo virtual.

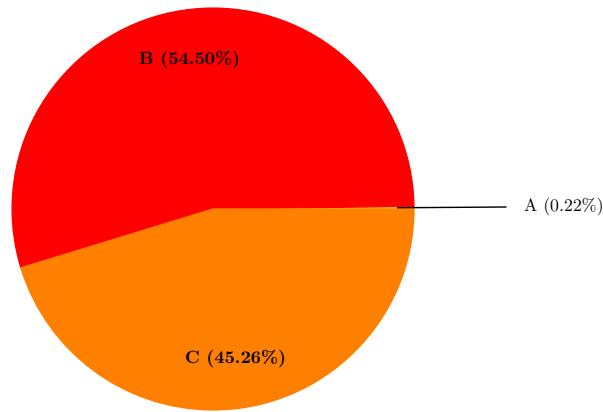


Figura 5.1 – Porcentajes de latencias totales por fase.

En la figura 5.2(a) se puede observar el histograma de las frecuencias de las latencias obtenidas al momentos de realizar las evaluaciones de las acciones del usuario. En el se obtiene una baja latencia, en donde su mayor frecuencia de latencia fue de 5 ms.

En la figura 5.2(b) se puede observar el histograma de las frecuencias de las latencias obtenidas del momento en el que se actualiza la información de las acciones del usuario. En el histograma se puede observar una gran variedad de latencias obtenidas, con un mínimo de 4 ms hasta un máximo de 38 ms, mientras que su mayor frecuencia de latencia fue de 6 ms.

En la figura 5.2(c) se puede observar el histograma de las frecuencias de las latencias obtenidas después del momento en el que se actualiza la información del usuario dentro del mundo virtual. En el se puede ver una variedad alta de latencias, en donde la mínima latencia obtenida es de 5 ms hasta un máximo de 5.6 segundos.

Dadas las figuras que se muestran se puede decir que la mayor latencia obtenida

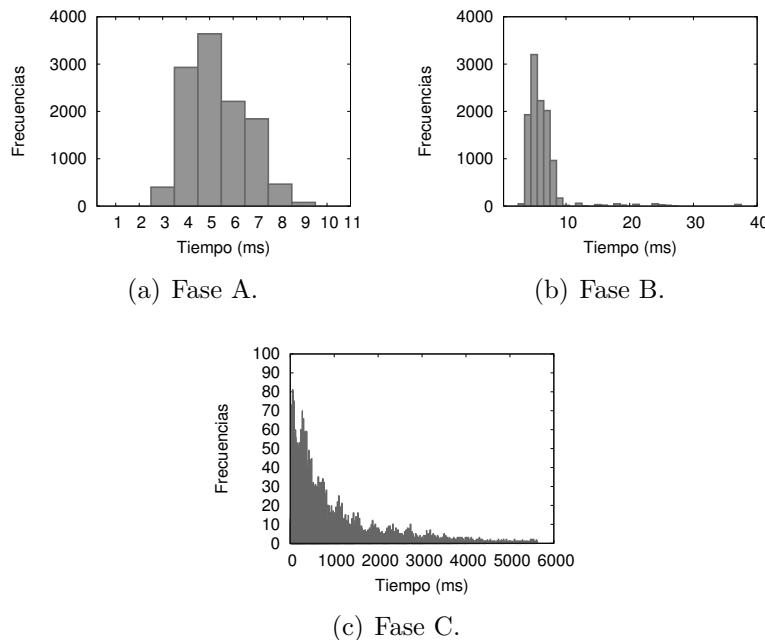


Figura 5.2 – Fases de las latencias del sistema de realidad virtual.

fue al momento que la información del usuario está siendo actualizada dentro del sistema de realidad virtual. Los causas a esta alta latencia se pueden deber a varios puntos entre ellos es a que recibe información del usuario con ruido debido a la inferferencias de los rayos infrarrojos del sol con el Kinect, otra causa puede ser debido a que el sistema de realidad virtual está actualizando información del usuario con una período preestablecido dentro de un script (parte del SDK de Oculus Rift).

La solución a este problema sería cambiar el área en la que se utiliza el sistema de realidad virtual tomando en consideración la interferencia que se puede causar por los rayos infrarrojos, además de cambiar la implementación de la actualización de la información de los pasos del usuario por una función independiente en el que se cambiará el tiempo establecido anteriormente por un tiempo más responsable acorde a límites de latencia aceptables.

En la fase B se observa un comportamiento diferente en comparación a las fases A y C. Esto se debe a que en la fase B se realiza la medición de la latencia

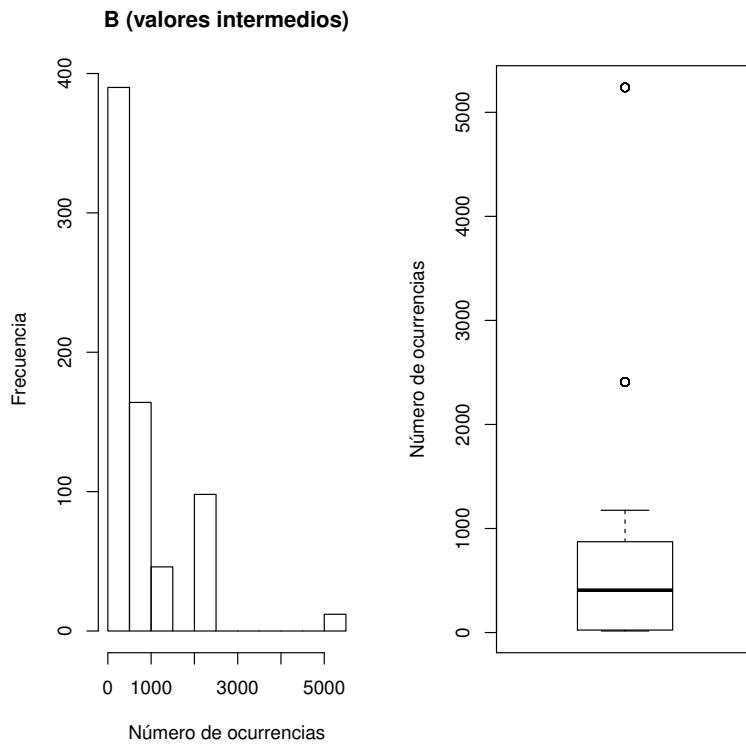


Figura 5.3 – Análisis de los valores intermedios de la latencia en fase B.

por medio de dos instancias de hilos: usuario y aplicación. Ambas instancias realizan mediciones de las latencias acorde a la secuencia definida por medio de *banderas*³ que ayudan a identificar una fase de otra. Sin embargo si estas instancias son ejecutadas en diferente orden por la calendarización de procesos (*scheduling* por su traducción al inglés) del sistema operativo, esté entra en conflicto⁴. Al llamarse una función dentro de una instancia que no corresponde al proceso definido y por lo tanto, no tomará los datos de la latencia hasta que se ejecute la función dentro de la instancia correspondiente a la secuencia definida.

Al analizar el comportamiento de los valores de la fase B se identificó manualmente que los datos son multimodales debido a que se identificaron tres grupos de los datos los cuales se pueden clasificar según su tamaño: altos, intermedios y bajos.

³Bandera es una variable que tiene como valor verdadero o falso, que son usadas en este trabajo para identificar una secuencia dentro de la aplicación.

⁴Orden de ejecución de funciones en Unity <http://docs.unity3d.com/Manual/ExecutionOrder.html>.

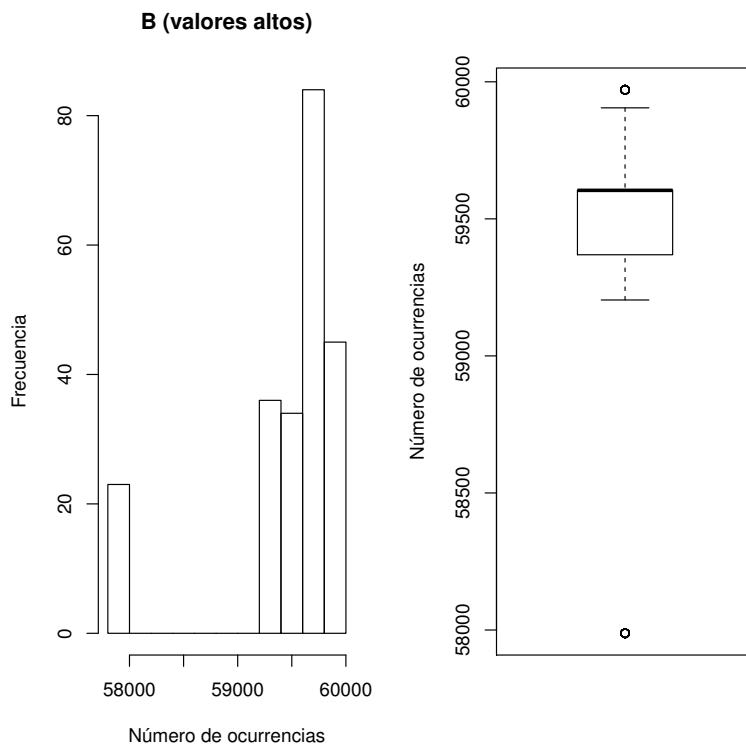


Figura 5.4 – Análisis de los valores altos de la latencia en fase B.

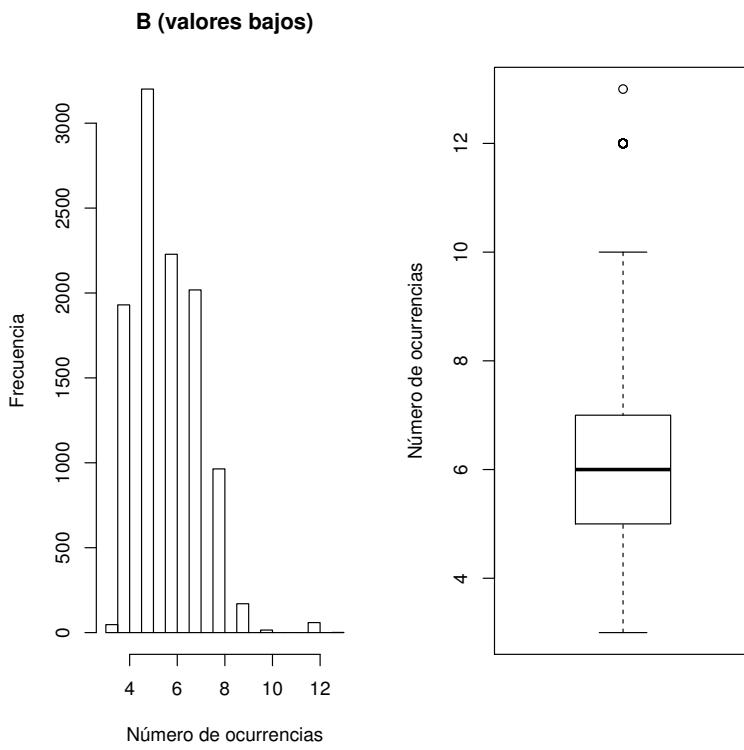


Figura 5.5 – Análisis de los valores bajos de la latencia en fase B.

Los tres diferentes grupos clasificados pueden ser observados en la figura 5.4, 5.3 y 5.5. En el histograma presentado se realizó un ajuste basándose en la moda de los datos, sin embargo como se puede observar en las figuras de los datos clasificados se tienen distintos grupos de datos.

Como anteriormente se había discutido, este comportamiento se debe al orden en el que se ejecutan las instancias de los hilos por la calendarización de procesos del sistema operativo. Sabiendo que la fase B es afectada por la calendarización de procesos, se planteó una hipótesis sobre los resultados obtenidos en la fase B. Los valores bajos son la respuesta deseada que implica que no hay ningún retraso considerable y que la calendarización de los procesos fue realizada de la manera deseada. Los valores intermedios y altos son una respuesta no deseada e implica que para que se haya ejecutado de manera deseada, se tuvieron que realizar varias iteraciones por la calendarización de los procesos.

5.2.3 EVALUACIÓN HEURÍSTICA

La evaluación heurística de usabilidad utilizada está basado en el trabajo de Sutcliffe y Gault [17] el cual presentan una lista de heurísticas inspiradas en el trabajo de Nielsen y Molich [11] con una modificación para cumplir la necesidad de hacer una evaluación de usabilidad de los ambientes virtuales y realidad virtual.

Para realizar las evaluaciones heurísticas de usabilidad se invitó a tres personas quienes han llevado una clase de interacción humano-computadora; esto debido a que uno de los conceptos básicos de la materia interacción humano-computadora es entender y realizar una evaluación heurística de algún trabajo relacionado. A cada uno de los evaluadores se les dio una lista de tareas a realizar utilizando el sistema de realidad virtual, esto fue con el fin de poder explorar el ambiente virtual y poder realizar la evaluación de las heurísticas.

En el cuadro E.1 que se encuentra en el apéndice E de la página 104 se pueden observar las evaluaciones que realizaron tres evaluadores. Esta evaluación fue realizada después de que los evaluadores hayan realizado un conjunto de tareas dentro del sistema de realidad virtual. El cuadro se compone por cuatro columnas: en la primer columna se tienen las heurísticas, en la segunda el número de evaluador, en la tercera columna a la clasificación dependiendo del valor que haya elegido (esto se discute en la sección de diseño) y en la última columna se presenta por parte de los evaluadores el problema que haya tenido.

Ahora se discutirán los resultados de la evaluación heurística de usabilidad que se presentan en el cuadro E.1 que se incluye en el apéndice E de la página 104. Se tomarán a consideración las heurísticas menos valoradas como resultado que asignaron los evaluadores. Las heurísticas menos valoradas por los evaluadores fueron: retroalimentación realista (5), navegación y soporte de orientación (7), salidas consistentes (9) y escencia de presencia (12).

Los problemas que se presentaron en la heurística de retroalimentación realista por parte de los evaluadores fueron sobre la reflección de los movimientos que realizaba el usuario en el mundo real, falta de interacción con más objetos y falta de calibración fueron los principales problemas. La reflección de los movimientos y la falta de calibración puede ser resuelto cambiando el área en la que se utiliza el sistema de realidad virtual, mientras que el problema de la falta de interacción con más objetos puede ir agregándose en otro diseño de algún escenario.

Según los evaluadores la navegación y soporte de orientación no tuvo resultados favorables debido a que se presentaron interferencias al utilizar el Kinect como dispositivo de retroalimentación. Lo anterior dando como resultado a que hubieron períodos en los que se presentaron interferencias de la proyección de rayos infrarrojos del Kinect, siendo que esta misma información corrupta del usuario es interpretada

por el programa que obtiene los datos del Kinect como movimientos inesperados ajenos a las acciones reales del usuario.

En la heurística de salidas consistentes se encontraron fallas del reconocimiento de los movimientos con Kinect por parte de los evaluadores. Las fallas que fueron encontradas por los evaluadores son debido a la interferencia de los rayos infrarrojos y a la lógica del reconocimiento de los movimientos. La solución a este problema encontrado sería mediante el cambio del área en el que se aplica el sistema de realidad virtual y el cambio del algoritmo de reconocimiento de movimientos.

La escencia de presencia no fue buena debido a que los evaluadores tuvieron problemas con el reconocimiento de los movimientos. Los evaluadores comentaron que la resolución del sistema de realidad virtual no era buena debido a que se ven los píxeles, sin embargo eso no detuvo la experiencia de sensación de inmersión dentro del mundo virtual. Una solución para la resolución sería tomar en consideración para una siguiente etapa otro dispositivo con una resolución más actualizada como lo es el nuevo paquete de desarrollo de Oculus Rift (DKV2) que ya cuenta con una resolución de 960 por 1080 píxeles por ojo.

En el cuadro E.2 que se encuentra en la página 109 de la clasificación de los problemas encontrados, los evaluadores resaltaron los problemas encontrados separados por temas como gráficas, presencia, interacción, características ambientales, controles y hardware. A continuación se verán los problemas de la clasificación de los problemas encontrados y sus conclusiones.

Gráficas En los problemas encontrados dentro de la sección de gráficas fueron la pobre resolución, teniendo en cuenta que esto es un proyecto inicial no es grave, sin embargo es una sugerencia a tomar a futuro en siguientes escenarios a crear.

Presencia El principal problema encontrado fue la detección de los pasos del usuario, como sugerencia sería realizar un cambio en el algoritmo para evitar repetir los mismos problemas presentados al realizar movimientos similares.

Interacción Un problema no grave pero si necesario es el cambio de tamaño de letra dentro de la interfaz del sistema de realidad virtual, debido a que la mayoría de los usuarios tuvieron problemas al leer las instrucciones.

Características ambientales El principal problema encontrado fue la velocidad de movimiento del usuario al ser detectado que camina en el mundo real. Como sugerencia sería realizar una detección independiente de un paso y calcular una velocidad en la que se realizan los pasos de dicho usuario.

Controles El problema encontrado es sobre la representación de las acciones que realiza en el mundo real con el mundo virtual, esto debido a problemas de reconocimiento de los movimientos del usuario y a la interferencia de los rayos infrarrojos en el área. Una solución a este problema sería establecer un área en el que no afecten los rayos infrarrojos al sistema de realidad virtual.

Hardware Se encontró un problema en el que el usuario no mencionó que tenía problemas de vista y que también debido a que faltó agregar en las instrucciones del uso del sistema de realidad virtual que el usuario puede cambiar de lentes del uso del Oculus Rift. Una sugerencia a este problema sería agregar a las instrucciones del uso del sistema de realidad virtual que existe la posibilidad de cambiar los lentes utilizados con el Oculus Rift y que en caso de no ver bien podrán ser cambiados por otros. Esto podría ser implementado con un caso de prueba poniendo unas letras para que el usuario pueda comprobar que observa de manera correcta un texto utilizando el Oculus Rift.

CAPÍTULO 6

CONCLUSIONES

En este trabajo se propuso realizar una aplicación de realidad virtual como base para realizar entrenamientos dado un conjunto de tareas propuestas. La hipótesis propuesta fue realizar un entrenamiento utilizando un sistema de realidad virtual, evaluando las acciones corporales de las personas que lo utilicen, así como el desempeño de las personas y sin problemas de usabilidad. Como objetivo principal se buscaba realizar una simulación de un entrenamiento por medio de la realidad virtual, asimismo tomando retroalimentación de las acciones corporales del usuario que irán siendo requeridas según las instrucciones del entrenamiento, además de evaluar las acciones tomadas y el tiempo de desempeño.

Los objetivos específicos dentro de este trabajo fueron ayudar a entrenar a personas para que puedan controlar una situación de peligro por medio de una serie de instrucciones que son presentadas dentro de la simulación, crear un sistema que ayude a evaluar el tiempo tomado y acciones, reducir el espacio utilizado contra una situación real y crear una interacción lo más natural posible y crear un escenario para ser usado en la simulación tomando en cuenta una actividad a realizar.

El marco de trabajo estuvo compuesto de una investigación de los antecedentes en donde se pueden encontrar los términos utilizados dentro de este trabajo. En la literatura existente se buscaron los trabajos existentes y se hizo un estudio compa-

rativo para identificar áreas de oportunidad. En la solución propuesta se propuso la metodología de software a utilizar y se planteó un conjunto de fases para desarrollar el sistema de realidad virtual. Las fases por las que se dividió fueron las siguientes: exploración de tecnologías, planeación de implementación, diseño de prototipo y implementación.

En el diseño de experimentos se propusieron tres experimentos, el primero fue el experimento del protocolo de pensamiento de en voz alta en donde se establece que el usuario realizará un conjunto de tareas establecidas utilizando el sistema de realidad virtual y al mismo tiempo, hablar en voz alta lo que está pensando. El segundo experimento fue el diseño de análisis de latencia en donde se buscó identificar la latencia del prototipo tomando un conjunto de muestras y diviéndolo en secciones para identificar cuellos de botella que puedan ocasionar una alta latencia. El tercer experimento fue el diseño de evaluación heurística siguiendo el trabajo propuesto Sutcliffe y Gault [17] en el que se evalúa la usabilidad del sistema de realidad virtual.

En la evaluación de experimentos se realizaron las evaluaciones de los experimentos propuestos. En la evaluación de protocolo de pensamiento en voz en alta fue presentada a tres usuarios. Los problemas que tuvieron los usuarios principalmente fue sobre el cambio de mensajes de instrucciones por medio del levantamiento de la mano derecha, problemas con interferencias con los rayos infrarrojos del sol (ocasionalmente) y problemas sobre la velocidad al momento de caminar el usuario en la vida real, reflejado en el mundo virtual. Luego se dieron conclusiones sobre posibles soluciones a este problema, el cual implica agregar usabilidad, cambio de espacio con menos luminosidad y cambio de algoritmo para la detección de paso.

En la evaluación de latencias se detectó que el sistema de realidad virtual tiene mayor latencia al momento en que recibe y se despliega la información recibida del sensor de retroalimentación hacia el mundo virtual. Las conclusiones a estos

problemas fueron que el sistema de realidad virtual deberá ser aplicado dentro de áreas específicas de tal forma para evitar interferencias con los rayos infrarrojos, también el algoritmo de la detección de los pasos del usuario deberá ser cambiado. También se detectó manualmente que los datos de la fase B son multimodales debido a que se identificaron tres grupos dentro de los datos. El comportamiento descubierto se debe al orden en el que se ejecutan las instancias de los hilos por la calendarización de procesos. Según la hipótesis planteada, los valores bajos son la respuesta deseada que implica que no hay ningún retraso considerable y que la calendarización de los procesos fue realizada de la manera deseada. Mientras que los valores intermedios y altos son una respuesta no deseada e implica que para que se haya ejecutado de manera deseada, se tuvieron que realizar varias iteraciones por la calendarización de los procesos. Como solución a este problema encontrado se tendría que realizar un análisis del comportamiento de las aplicaciones en Unity e intentar cambiar la manera en la que se ejecutan los hilos en el sistema de realidad virtual.

Por medio de la evaluación heurística se identificó que el usuario no tuvo síntomas de mareo ocasionadas por la latencia obtenida en la evaluación de latencia al momento en el que el evaluador utilizó la aplicación de realidad virtual, sin embargo se identificaron problemas al realizar ciertas acciones por distintos evaluadores. Los problemas encontrados fueron relacionados a la detección de movimiento del usuario, estos mayormente ocasionados por una interferencia por los rayos infrarrojos. Se identificó también que si los usuarios realizaban una acción parecida a la solicitada, esta sería aceptada como el movimiento requerido. Se tuvieron problemas de usabilidad debido a que el usuario no sabía si el sistema de realidad virtual lo seguía detectando, debido a que en el mundo virtual el usuario se detiene cuando no es detectado en la vida real. Además de que el sistema no proporciona retroalimentación de detección de movimiento por medio de un ícono que señale el movimiento detectado. También se identificó que la velocidad del movimiento de las piernas no

es reflejada en el mundo virtual como en el mundo real. Se propuso que se resolverían estos problemas por medio del cambio del área de aplicación del sistema de realidad virtual tomando en consideración la luminosidad para evitar interferencias con los rayos infrarrojos, cambiar el algoritmo de detección de las piernas y el tiempo por el que se actualiza la información del movimiento de los pies del usuario.

Dentro de la clasificación de los problemas encontrados por parte de los evaluadores se identificaron más problemas técnicos como la pobre resolución, la opción de ofrecerles a los usuarios el cambio de lentes de los Oculus Rift y cambio de tamaño de las letras de las instrucciones a más grandes. Como solución puede ser la sustitución del hardware por un sistema más actualizado, cambiar el tamaño de la letras de las instrucciones del sistema de realidad virtual y agregar dentro de las instrucciones que existe la posibilidad de cambio de lentes.

En trabajo a futuro sería extender las acciones evaluadas, ya que en este trabajo solamente se pueden evaluar acciones como mover adelante, derecha e izquierda. También como trabajo a futuro sería realizar un entorno virtual más complejo como una situación en donde se tengan que realizar tareas más complejas como desactivar granadas, apagar fuego en un edificio, situaciones de asalto, etcétera, todo esto se puede realizar por medio de una serie de tareas compuestas en un cierto orden. Agregando tecnologías emergentes como el MYO¹ el cual es un sensor muscular en forma de brazalete el cual ayuda a determinar la posición exacta y movimiento de la mano, esto podría enriquecer la experiencia de usuario de la realidad virtual en comparación a otros estudios en donde utilizaron guantes con sensores que flexibilidad para calcular el movimiento de cada dedo de la mano y evaluar su posición actual. También es posible reemplazar la funcionalidad de localización del Kinect por un sistema de sensores que realizan una detección de la localización del usuario por medio de una triangulación en el área donde se encuentra el usuario.

¹Sitio oficial de MYO <https://www.thalmic.com/en/myo>.

BIBLIOGRAFÍA

- [1] Michael Antonov, Nate Mitchell, Andrew Reisseand Lee Cooper, Steve La-Valle y Max Katsev. *Oculus SDK Overview SDK Version 0.2.5.* Oculus VR, Inc, 2013. http://static.oculusvr.com/sdk-downloads/documents/Oculus_SDK_Overview.pdf.
- [2] Kent Beck y Cynthia Andres. *Extreme Programming Explained: Embrace Change*, volúmen 2. Addison-Wesley, Indianapolis, IN, EUA, 2004.
- [3] Sue Blackman. *Beginning 3D Game Development with Unity 4: All-in-one, multiplatform game development*, volúmen 2. Apress, Nueva York, NY, EUA, 2013.
- [4] Michael Chau, Ada Wong, Minhong Wang, Songnia Lai, Kristal W. Y. Chan, Tim M. H. Li, Debbie Chu, Ian K. W. Chan y Wai-Ki Sung. Using 3D virtual environments to facilitate students in constructivist learning. *Decision Support Systems*, 56:115–121, 2013.
- [5] Shel Israel. Why Apple bought PrimeSense. Forbes, 2013. <http://www.forbes.com/sites/shelisrael/2013/11/25/why-would-apple-buy-primesense/>.
- [6] Eric Lengyel. *Matemáticas Para Videojuegos En 3D*, volúmen 2. Cengage Learning, Santa Fe, D.F., México, 2013.
- [7] Benjamin Lok, Samir Naik, Mary C. Whitton y Frederick P. Brooks Jr. Effects of

- Handling Real Objects and Self-Avatar Fidelity on Cognitive Task Performance and Sense of Presence in Virtual Environments. *Presence*, 12(6):615–628, 2003.
- [8] David Meyer, Achim Zeileis, Kurt Hornik, Florian Gerber y Michael Friendly. *Support Functions and Datasets for Venables and Ripley's MASS*, 2014. <http://cran.r-project.org/web/packages/MASS/MASS.pdf>.
- [9] David Meyer, Achim Zeileis, Kurt Hornik, Florian Gerber y Michael Friendly. *Visualizing Categorical Data*, 2014. <http://cran.r-project.org/web/packages/vcd/vcd.pdf>.
- [10] Jakob Nielsen. *Usability Engineering*, volúmen 1. Morgan Kaufmann, San Francisco, CA, EUA, 1993.
- [11] Jakob Nielsen y Rolf Molich. Heuristic Evaluation of User Interfaces. En *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, páginas 249–256. ACM, Nueva York, NY, EUA, 1990. ISBN 0-201-50932-6.
- [12] Vito Ricci. *Fitting Distributions With R*, 2005. <http://cran.r-project.org/doc/contrib/Ricci-distributions-en.pdf>.
- [13] Jonathan P. San Diego, Margaret J. Cox, Barry F.A. Quinn, Jonathan Tim Newton, Avijit Banerjee y Mark Woolford. Researching haptics in higher education: The complexity of developing haptics virtual learning systems and evaluating its impact on students' learning. *Computers & Education*, 59(1):156–166, 2012.
- [14] William R. Sherman y Alan B. Craig. *Understanding Virtual Interface: Interface, Application, Design*, volúmen 2. Morgan Kauffman, San Francisco, CA, EUA, 2003.

- [15] Peter Shirley, Michael Ashikhmin y Steve Marschner. *Fundamentals of Computer Graphics*, volúmen 3. A K Peters/CRC Press, Boca Raton, FL, EUA, 2009.
- [16] Sharon A. Stansfield, Daniel Shawver, Annette Sobel, Monica Prasad y Lydia Tapia. Design and Implementation of a Virtual Reality System and Its Application to Training Medical First Responders. *Presence*, 9(6):524–556, 2000.
- [17] Alistair Sutcliffe y Brian Gault. Heuristic evaluation of virtual reality applications. *Interacting with Computers*, 16:831–849, 2004.
- [18] Andrew S. Tanenbaum. *Computer Networks*, volúmen 5. Prentice Hall, Upper Saddle River, Nueva Jersey, C.P. 07458, EUA, 2010.
- [19] Etienne Van Wyk y Ruth de Villiers. Virtual reality training applications for the mining industry. En Alexandre Hardy, Patrick Marais, Stephen N. Spencer, James E. Gain, y Wolfgang Straßer, editores, *Afrigraph*, páginas 53–63. ACM, Nueva York, NY, EUA, 2009.
- [20] Zhen Xu, Xinzhrng Lu, Hong Guan, Chi Chen y Aizhu Z. Ren. Simulation of Emergency Evacuation in Virtual Reality. *Tsinghua Science and Technology*, 13:674–680, 2008.
- [21] Richard Yao, Tom Heath, Aaron Davies, Tom Forsyth, Nate Mitchell y Perry Hoberman. *Oculus VR Best Practices Guide*. Oculus VR, Inc, 2014. <http://static.oculusvr.com/sdk-downloads/documents/OculusBestPractices.pdf>.

APÉNDICE A

HISTORIA DEL USUARIO DEL SISTEMA DE REALIDAD VIRTUAL

Esta es la historia del papel del usuario del uso del sistema de realidad virtual.

Miguel es un estudiante de preparatoria y va a utilizar un sistema de realidad virtual. Miguel entra a un espacio seleccionado por el encargado de la asistencia del uso del sistema de realidad virtual. El asistente le muestra a Miguel una hoja con instrucciones sobre el uso del sistema de realidad virtual, para que se familiarice con el sistema de realidad virtual. Después se le pide a Miguel que esté en un área señalada, el asistente procede a entregarle el dispositivo Oculus Rift a Miguel.

Luego el asistente le pregunta a Miguel si necesita ayuda con el ajuste del dispositivo Oculus Rift, en caso de ser necesaria la ayuda el asistente se lo proveerá. Una vez que tenga ajustado Miguel el dispositivo de Oculus Rift, el asistente procederá a presionar el botón de comenzar dentro de la interfaz de la aplicación. Esto va ocasionar que Miguel pueda ver un texto en el dispositivo de Oculus Rift, el cual dice lo siguiente “Detectando usuario...”. En éste se espera que Miguel sea detectado por el Kinect, el cual es un dispositivo que se encuentra en una posición adecuada para

que sea detectado Miguel.

Después de que haya sido detectado Miguel, él verá un texto que dice “Levante la mano derecha para continuar...”, con esto el sistema de realidad virtual espera que Miguel levante la mano derecha para continuar. Al levantar la mano derecha, se comenzará a desplegar un modelo 3D de un cuarto en el que se muestran instrucciones de la actividad a realizar las cuales son las siguientes “Visite las siguientes áreas en el orden y color de los siguientes colores.” y debajo de esas instrucciones se muestra un orden aleatorio de colores el cual Miguel deberá visitar.

Miguel tendrá que utilizar el torso para determinar la dirección por la cual se dirigirá dentro el sistema de realidad virtual, esto también funciona con el movimiento del cuello. Para realizar movimiento de un lugar a otro tendrá que mover las piernas Miguel tal como lo hace al subir escaleras permaneciendo en el área señalada. En caso de ir a un color distinto al orden señalado, éste volverá a la posición inicial.

Al terminar de visitar los colores se le mostrará a Miguel el tiempo que tardó en realizar la actividad y con esto habrá terminado la actividad del sistema de realidad virtual. En caso de querer reiniciar la actividad, el usuario tendrá que levantar la mano derecha.

APÉNDICE B

DIAGRAMAS FUNCIONALES DEL SISTEMA DE REALIDAD VIRTUAL

En la figura B.1 se puede observar el diagrama de estados máquina del mundo virtual; en el capítulo 4 de solución propuesta se explica cada uno de los estados y cómo pasa de una transición a otra.

Los diagramas de secuencia que se pueden observar en la figura B.2 y B.3 en las cuales se puede observar la secuencia del sistema de realidad virtual. Dicho proceso es explicado en el capítulo de solución propuesta.

Por último, en la figura B.4 se presenta el digrama de flujo de la aplicación que obtiene información del usuario. Este diagrama también es explicado en el capítulo de solución propuesta.

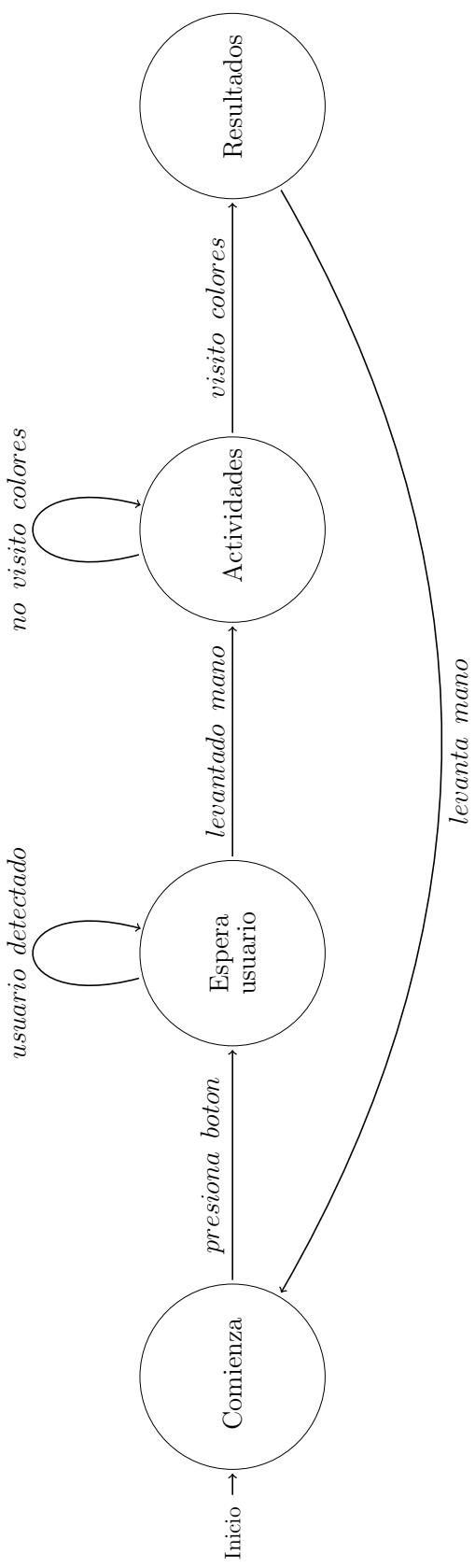


Figura B.1 – Diagrama de estados máquina del mundo virtual.

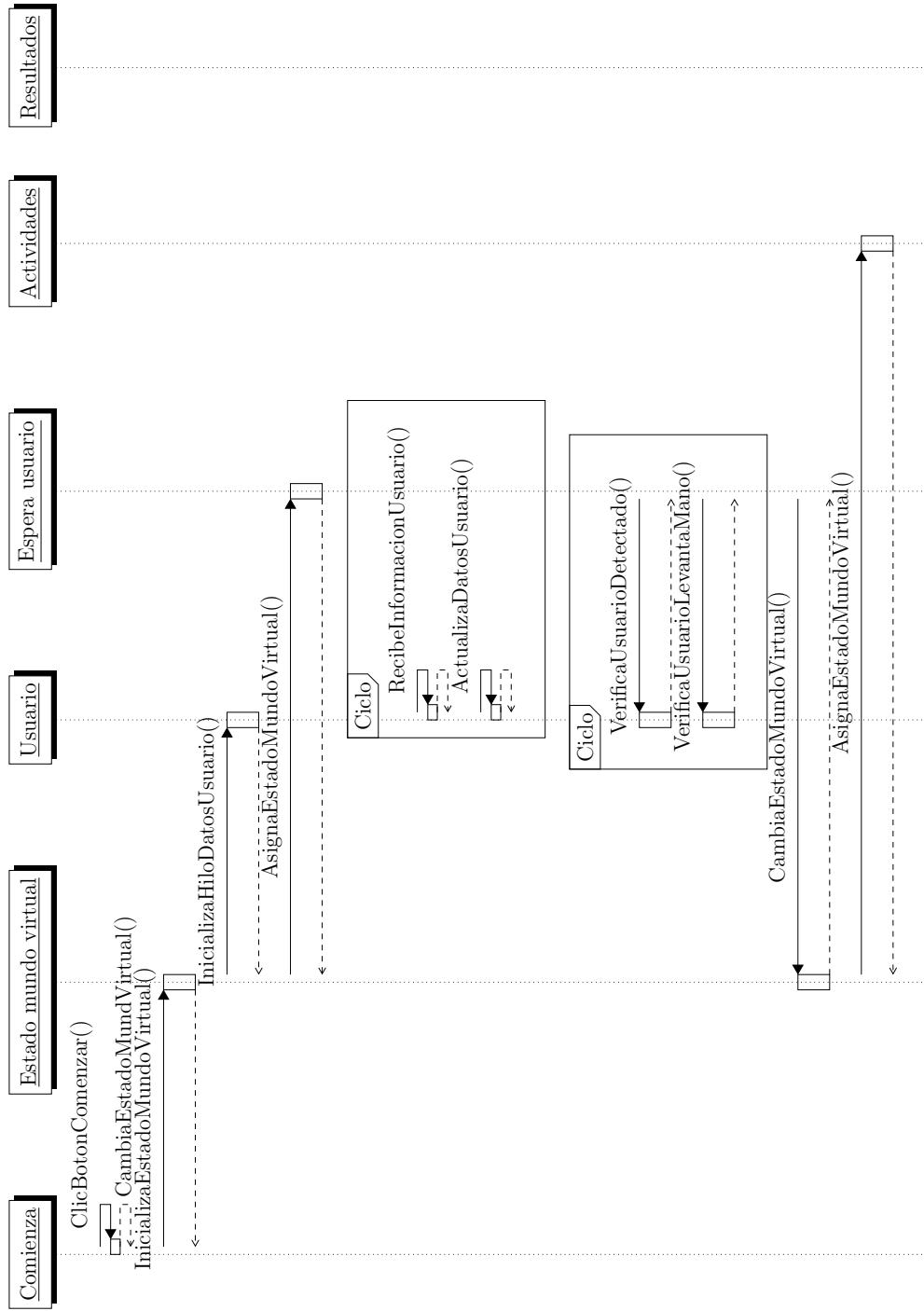


Figura B.2 – Diagrama de secuencias del mundo virtual (parte 1).

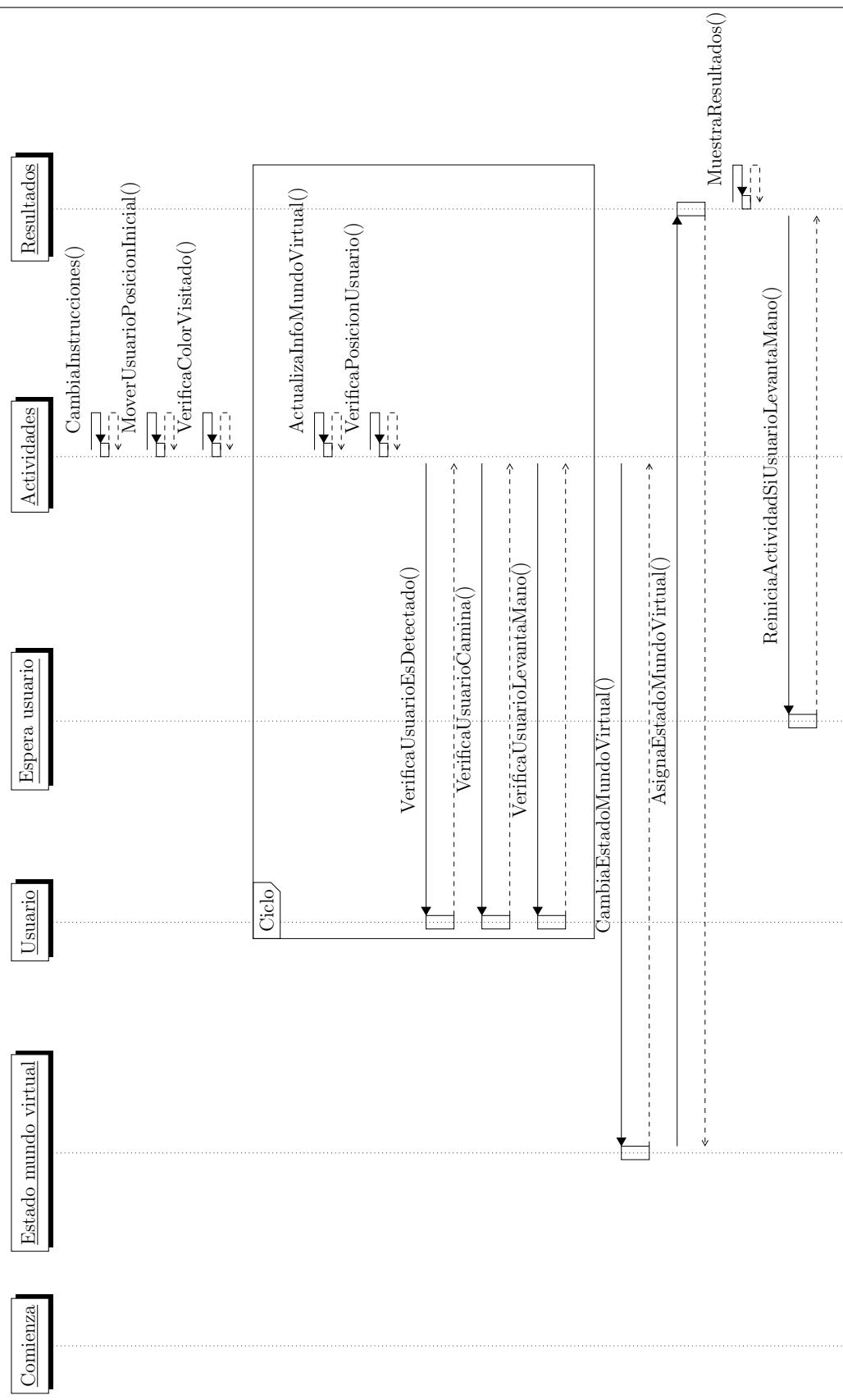


Figura B.3 – Diagrama de secuencias del mundo virtual (parte 2).

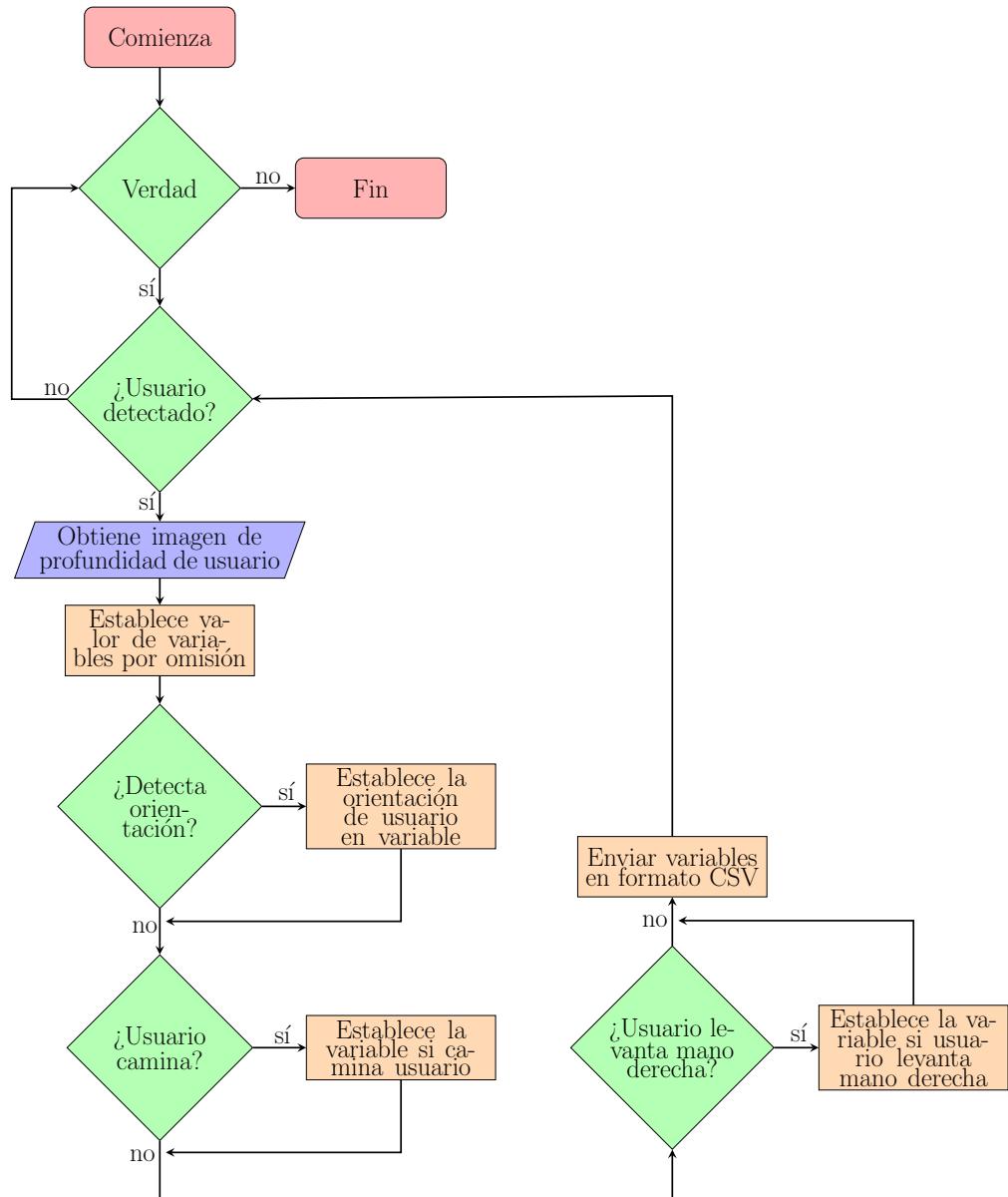


Figura B.4 – Diagrama de flujo de la aplicación que obtiene información del usuario.

APÉNDICE C

INSTRUCCIONES DE LA ACTIVIDAD DEL SISTEMA DE REALIDAD VIRTUAL

Para motivos de la aplicación del sistema de realidad virtual se realizó un instructivo en donde se explica el uso del sistema de realidad virtual. El instructivo inicia con el siguiente párrafo: “Leer instrucciones y familiarizarse con ellas debido a que serán requeridas según el orden en el que fueron puestas.”. Luego se muestra un listado de instrucciones que serán requeridas según el orden presentado. El listado de las instrucciones es el siguiente:

- Posicionarse en el punto determinado por su asistente.
- Colocarse dispositivo Oculus Rift, en caso de requerir asistencia para ajustar u otra actividad favor de preguntar a asistente.
- Tendrá que ser detectado por un sensor, así que cuando vea el texto “Detectando usuario...” dentro del dispositivo de Oculus Rift trate de despegar brazos del cuerpo e intente abrir piernas un poco para que el sensor pueda detectarlos.
- Cuando vea el texto “Levantar mano derecha para continuar...” dentro del dispositivo de Oculus Rift será necesario que levante la mano derecha para continuar con el juego.

- Dentro del juego se encuentra las instrucciones con este texto “Visite cada una de las áreas correspondientes según su color y orden en el que son presentados:” y debajo del mismo se encontrará una lista de colores que cada vez cambia y de los cuales tendrán que ser visitados, esto mismo dentro del dispositivo de Oculus Rift.
- Las instrucciones pueden ser removidas levantando la mano derecha, esto oca-sionará que aparezca una ayuda de los colores faltantes a visitar.
- Para visitar los colores tendrá que mover su torso para dirigir la dirección a la que se desea ir, esto también funciona con el movimiento del cuello. Para trasladarse tendrá que hacer movimientos como si estuviera caminando pero estos tienen que ser de manera estática dentro del área señalada por el asistente.
- Al llegar a un color no requerido en las instrucciones será movido a una posición inicial, en caso de visitar todos los colores terminará el juego y se le mostrará el tiempo que tardó en terminar el juego.
- Cuando se le muestra el tiempo que tardó en terminar el juego podrá levantar la mano derecha para que vuelva a reiniciar el sistema de realidad virtual desde el inicio.

Para futuras implementaciones del sistema de realidad virtual se busca que se pueda implementar el instructivo dentro del mundo virtual en donde se muestre un video, audio o en donde el usuario pueda ser guiado por un avatar realizando un uso básico del control del sistema de realidad virtual.

APÉNDICE D

PRUEBAS DE BONDAD DE AJUSTE DE LAS LATENCIAS

En este apéndice se hace el desarrollo de las pruebas de bondad de ajuste (*goodness of fit* por su traducción al inglés) para las fases A, B y C.

Las pruebas de bondad son realizadas con el lenguaje de programación llamado R¹, en donde provee funciones para realizar estadísticas y gráficas computacionales.

La prueba de bondad realizada para la distribución de Poisson de la cual se presentaron en las fases A y B que se pueden observar en la figura D.1 y la figura D.2, mientras que en la fase C se utilizó una prueba de bondad distinta debido a que presenta una distribución exponencial, tal como se presenta en la figura D.3. Estas pruebas de bondad están basadas en el trabajo de Ricci [12], en donde demuestra distintas pruebas de bondad para diferentes distribuciones utilizando el lenguaje de programación R y las librerías *vcd* [9] y *MASS* [8].

A continuación se presentan como programas 1 y 2 el código en R que realiza la prueba de bondad para la distribución de Poisson. En el programa 3 se presenta el código en R que realiza la prueba de bondad para la distribución exponencial.

¹Sitio de el lenguaje script R <http://www.r-project.org/>.

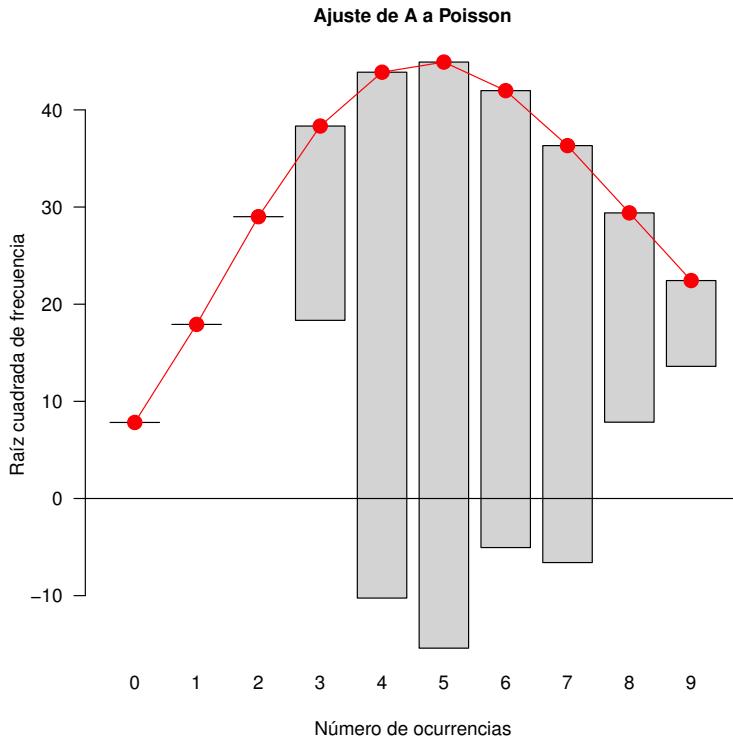


Figura D.1 – Ajuste de los valores de la fase A a la distribución de Poisson.

Programa 1 Código en R para prueba de bondad de ajuste de la fase A a Poisson.

```
library(vcd)
library(MASS)
gfa = goodfit(a, type = "poisson", method = "MinChisq")
png("ajuste_a.png", width=500, height=500)
plot(gfa, main="Ajuste de A a Poisson")
graphics.off()
```

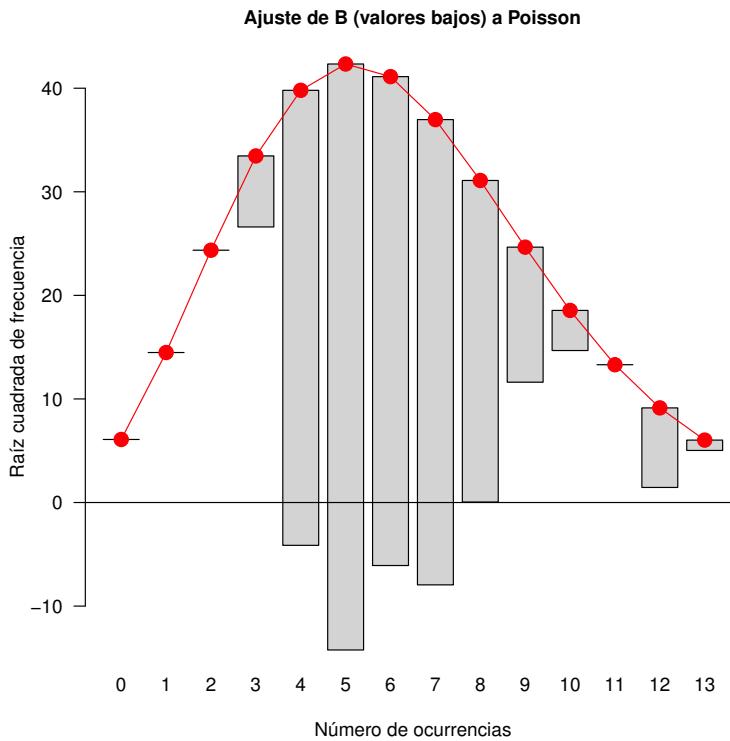


Figura D.2 – Ajuste de los valores de la fase B bajos a la distribución de Poisson.

Programa 2 Código en R para prueba de bondad de ajuste de la fase B a Poisson.

```
library(vcd)
library(MASS)
b_bajo = b[b < 15]
gfb_bajo = goodfit(b_bajo, type = "poisson", method = "MinChisq")
summary(gfb_bajo)
png("analisis_b_bajo.png", width=500, height=500)
par(mfrow=c(1,2))
hist(b_bajo, main="B (valores bajos)")
boxplot(b_bajo, main="")
png("ajuste_b_bajo.png", width=500, height=500)
plot(gfb_bajo, main="Ajuste de B (valores bajos) a Poisson")
graphics.off()
```

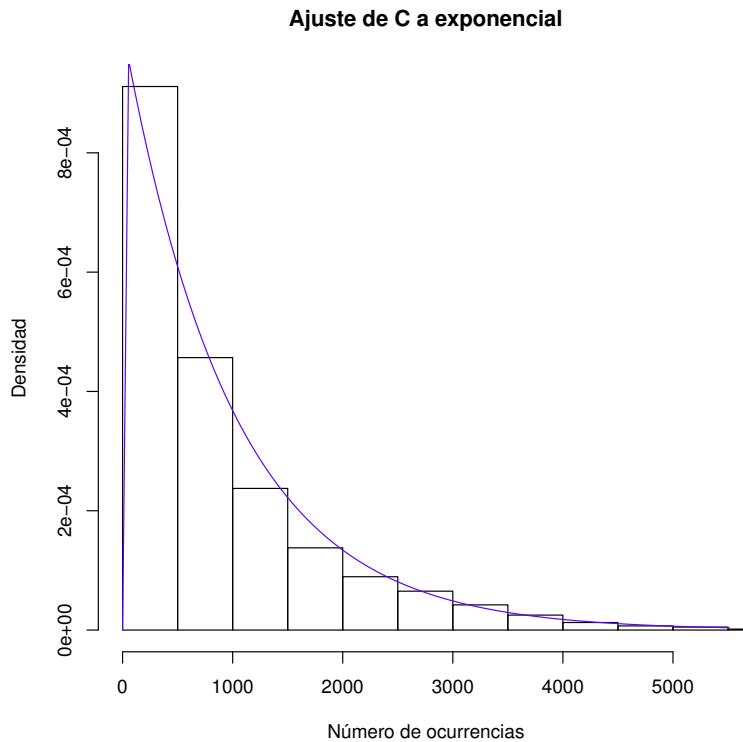


Figura D.3 – Ajuste de los valores de la fase C a la distribución exponencial.

Programa 3 Código en R para prueba de bondad de ajuste de la fase C a exponencial.

```
library(vcd)
library(MASS)
fc = fitdistr(c, "exponential")
ks.test(c, "pexp", fc$estimate)
png("ajuste_c.png", width=500, height=500)
hist(c, freq=FALSE, breaks=80, xlim=(c(0, quantile(c, 0.995))),
main="Ajuste de C a exponencial")
curve(dexp(x, rate=fc$estimate), col="blue", add=TRUE)
graphics.off()
```

APÉNDICE E

RESULTADOS DE LA EVALUACIÓN HEURÍSTICA

A continuación se presentan los resultados de la evaluación heurística de usabilidad, tal como fue propuesto en el capítulo de experimentos. Las clasificaciones se encuentran ponderadas de *1* a *4* donde *1* es clasificada como *malo* y *4* como *excelente*, en situaciones en donde el evaluador determine que no se aplica dicha evaluación debido a que el sistema virtual no proporciona dicha funcionalidad a evaluar se coloca *NA* para expresar que no aplica.

Cuadro E.1 – Resultado de la evaluación de las heurísticas.

1. Atado natural		
Evaluadores	Calificación	Problema
Evaluador # 1	3	La iluminación de la pantalla es distinta a la de un entorno natural, aun así no afecta mucho ya que la sensación de estar en otro mundo es muy fuerte.
Evaluador # 2	3	Bueno, el punto anterior es muy difícil de conseguir, yo creo que es aceptable.

Evaluador # 3	3	Existe un retraso en los movimiento, además de la necesidad de calibrar la posición del sujeto de manera frecuente debido al limitado campo sensorial.
---------------	---	--

2. Compatibilidad con las tareas y dominio del usuario

Evaluadores	Calificación	Problema
Evaluador # 1	4	Los objetos en este caso los muros, el suelo, etc., se comportan como en el mundo real, al momento de desplazarte por el entorno, están fijos y puedes sentir que son reales porque al cambiar tu posición también cambia su aspecto (como en el mundo real).
Evaluador # 2	3	Es intuitivo, hasta cierto punto, porque las instrucciones en texto son difíciles de leer.
Evaluador # 3	4	Las acciones producen los efectos pensados, solo que con un corto retraso.

3. Expresión natural de acción

Evaluadores	Calificación	Problema
Evaluador # 1	4	El entorno se puede explorar de manera similar a como se exploraría una habitación en el mundo real, salvo que hay un par de problemas al dar los pasos, si los das rápido, la velocidad no aumenta, ni tampoco la distancia al simular un paso mas alto.

Evaluador #2	2	El sensor de Kinect estaba mal sincronizado, había varios problemas dentro de la detección del movimiento.
Evaluador # 3	3	Hay una sensación de inmersión buena, ya que tus sentidos se enfocan a realizar acciones dentro de ese ambiente virtual.

4. Estrecha coordinación de acción y representación

Evaluadores	Calificación	Problema
Evaluador # 1	3	A pesar que si se veía afectado el tiempo de respuesta (desconozco la cantidad de tiempo) no percibí ningún tipo de mareo o malestar físico.
Evaluador # 2	3	Aunque habían retrasos, no había problemas de mareo.
Evaluador # 3	3	Hay una estrecha relación entre las acciones tomadas en la vida real y las reflejadas en el ambiente virtual, siendo afectadas únicamente por la lenta velocidad de respuesta.

5. Retroalimentación realista

Evaluadores	Calificación	Problema
Evaluador # 1	3	Los movimientos que realizaba en base a las instrucciones se veían claramente reflejados, si tu avanzabas la escena virtual se desplazaba en sentido contrario de acuerdo a como se percibe físicamente en la realidad.

Evaluador # 2	3	No se logran apreciar todas las leyes de la física, tal vez hace falta más objetos dentro del proyecto.
Evaluador # 3	2	Falta establecer límites mínimos y máximos necesarios para la interpretación de movimientos al realizar acciones básicas.

6. Puntos de vista leales

Evaluadores	Calificación	Problema
Evaluador # 1	4	No había ningún retraso en la renderización de los gráficos.
Evaluador # 2	4	El dispositivo Oculus Rift hacía muy buen trabajo dentro de unity.
Evaluador # 3	4	Muy buena la representación del mundo virtual representa una fiel copia de los movimientos de la cabeza.

7. Navegación y soporte de orientación

Evaluadores	Calificación	Problema
Evaluador # 1	3	El ambiente virtual es fácilmente reconocible y la orientación no es ningún problema para el usuario.
Evaluador # 2	3	Era complicado mantener la ubicación de uno mismo.
Evaluador # 3	2	Debido a la falta de límites establecidos para determinar cuando un movimiento se termina de efectuar o apenas inicia, se producen muchas acciones inesperadas o no deseadas.

8. Entrada clara y puntos de salida		
Evaluadores	Calificación	Problema
Evaluador # 1	3	En las instrucciones provistas vía gráfica se mostraba claramente cuando la realidad virtual estaba llevándose acabo y cada vez que terminaba la participación del usuario en el ambiente.
Evaluador # 2	4	Estaban correctamente señalados.
Evaluador # 3	4	Claramente establecido y reconocible.
9. Salidas consistentes		
Evaluadores	Calificación	Problema
Evaluador # 1	4	El diseño cumple el objetivo sin ningún problema, y es fácilmente realizable.
Evaluador # 2	3	Se cumple el cometido de manera aceptable, un par de problemas mínimos nada más.
Evaluador # 3	1	Se compromete la integridad del diseño al ser muy fácil el identificar los patrones en los que falla el sistema encargado de interpretar los movimientos.
10. Soporte para aprender		
Evaluadores	Calificación	Problema
Evaluador # 1	3	Los objetivos estaban claramente señalizados por medio de colores.
Evaluador # 2	4	El reinicio de la posición es muy buen mecanismo de auto aprendizaje.
Evaluador # 3	3	Diseño intuitivo y claro.
11. Turnos claros		

Evaluadores	Calificación	Problema
Evaluador # 1	4	El sistema mostraba una interfaz gráfica que mostraba tu turno y estado.
Evaluador # 2	4	Señalado apropiadamente.
Evaluador # 3	4	Marca bien puntos de inicio y termino, entradas y salidas del mundo virtual.

12. Escencia de presencia

Evaluadores	Calificación	Problema
Evaluador # 1	3	Las texturas pueden afectar un poco, pero la sensación de estar en un mundo es completamente perceptible.
Evaluador # 2	3	Lo único que afecta la sensación natural fueron los problemas con Kinect y la mala coordinación con las piernas.
Evaluador # 3	2	Percepción visual buena, percepción corporal mala debido al retraso y falta de ajustes.

En el cuadro E.2 se muestra clasificación de los problemas encontrados, los evaluadores resaltaron los problemas encontrados separados por temas como gráficas, presencia, interacción, características ambientales, controles y hardware.

Cuadro E.2 – Clasificación de los problemas encontrados.

Gráficas			
Evaluadores	Descripción	Clasificación	Cambio de diseño

Evaluador # 1	Pobre resolución.	Inconveniente	Se puede cambiar, ya que el problema era la resolución de la pantalla.
Evaluador # 2	Son apropiadas para un proyecto inicial.	Inconveniente	Espero en etapas más avanzadas se encuentren más objetos y texturas así también como un mapa más grande.
Evaluador # 3	Pobre detalle en el ambiente gráfico y baja resolución, haciendo que la interpretación de caracteres se dificulte.	Inconveniente	Trabajar con una resolución mas alta e incluir mas detalle en las gráficas.
Presencia			
Evaluadores	Descripción	Clasificación	Cambio de diseño

Evaluador # 1	El único problema en la navegación es cuando das pasos rápidos la velocidad no aumenta, ni la distancia. El uso del sensor hace que el sistema de una sensación de realidad agregada.	Inconveniente	Cambiar la forma en la que funcionan los pasos, si los pasos son constantes la velocidad podría aumentar pero la distancia entre pasos sería menor, eso le daría más sensación de realidad.
Evaluador # 2	La falta de interacción con las piernas (de detección).	Molesto	Mejorar la detección.
Evaluador # 3	Falta de fidelidad en el sistema al interpretar movimiento hechos por el usuario.	Molesto	Incluir límites mínimos y máximos para identificar con éxito cuando un movimiento apenas inicia y cuando termina completamente.
Interacción			
Evaluadores	Descripción	Clasificación	Cambio de diseño
Evaluador # 1	Todo estaba bastante claro en este sentido.	NA	NA
Evaluador # 2	Letras muy pequeñas en las instrucciones.	Inconveniente	Hacer más grande las letras.

Evaluador # 3	La interacción con objetos en el ambiente virtual en veces se vuelve algo problemática debido a la posición.	Inconveniente	Mejorar el sistema de coordenadas.
---------------	--	---------------	------------------------------------

Características ambientes

Evaluadores	Descripción	Clasificación	Cambio de diseño
Evaluador # 1	El problema que tal vez existe es cuando simulabas la posición de un paso, pero permanecías con el pie en el aire, el sistema lo interpretaba como paso continuo y nunca dejaba de desplazarte.	Distraído	Hacer que cuando levantes el pie solamente de un paso.
Evaluador # 2	NA	NA	NA
Evaluador # 3	NA	NA	NA

Controles

Evaluadores	Descripción	Clasificación	Cambio de diseño
Evaluador # 1	NA	NA	NA
Evaluador # 2	NA	NA	NA

Evaluador # 3	Debido a la necesidad de primero calibrar el sensor de movimiento con el usuario, algunas tareas se vuelven confusas al intentarlas por primera vez.	Distraído	Aumentar rápido en el reconocimiento del usuario y aumentar el campo sensorial efectivo.
---------------	--	-----------	--

Hardware

Evaluadores	Descripción	Clasificación	Cambio de diseño
Evaluador # 1	NA	NA	NA
Evaluador # 2	NA	NA	NA
Evaluador # 3	Al utilizar lentes para enfocar muy cerca de los ojos, se crea la necesidad de ajustarlos de acuerdo a las capacidades diferentes de los usuarios.	Inconveniente	Darle la opción al usuario de utilizar lentes de distintos enfoques, adecuados para cada tipo de persona.

FICHA AUTOBIOGRÁFICA

Gerardo Saúl Gausin Valle

Candidato para el grado de Ingeniero en Tecnología de Software

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

**APLICACIONES DE REALIDAD VIRTUAL EN SISTEMAS
DE ENTRENAMIENTO PARA TAREAS FÍSICAS**

Nací en la ciudad de Monterrey, Nuevo León el 18 de junio de 1992. Mis padres son Juan Gerardo Gausin Trejo y Griselda Valle Luna. Soy estudiante de la Facultad de Ingeniería Mecánica y Eléctrica de la Universidad Autónoma de Nuevo León como Ingeniero en Tecnología de Software. La preparatoria por la que egresé fue la Preparatoria 15 Florida. Entre mis pasatiempos son conocer nuevas tecnologías y aprender de ellas.