



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA Y
TELECOMUNICACIONES

TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
ENTRETENIMIENTO LÚDICO BASADO EN IDENTIFICACIÓN
DE MOVIMIENTOS DE MANOS A DISTANCIAS CORTAS
USANDO EL SENSOR LEAP MOTION CONTROLLER

AUTOR: PAZMIÑO MOYA VINICIO DAVID

DIRECTOR: ING. ALULEMA FLORES DARWIN OMAR MGS.

SANGOLQUÍ

2017



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que el trabajo de titulación "**DISEÑO E IMPLEMENTACION DE UN SISTEMA DE ENTRETENIMIENTO LÚDICO BASADO EN IDENTIFICACIÓN DE MOVIMIENTOS DE MANOS A DISTANCIAS CORTAS USANDO EL SENSOR LEAP MOTION CONTROLLER**" realizado por el señor **PAZMIÑO MOYA VINICIO DAVID**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos, y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **PAZMIÑO MOYA VINICIO DAVID** para que lo sustente públicamente.

Sangolquí, 9 de agosto de 2017

Ing. Darwin Alulema MGS.
DIRECTOR



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES

AUTORÍA DE RESPONSABILIDAD

Yo, **PAZMIÑO MOYA VINICIO DAVID**, con cedula de identidad N° 1719989889, declaro que este trabajo de titulación **“DISEÑO E IMPLEMENTACION DE UN SISTEMA DE ENTRETENIMIENTO LÚDICO BASADO EN IDENTIFICACIÓN DE MOVIMIENTOS DE MANOS A DISTANCIAS CORTAS USANDO EL SENSOR LEAP MOTION CONTROLLER”** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad, y alcance de la investigación mencionada.

Sangolquí, 9 de agosto de 2017

Vinicio David Pazmiño Moya

C.C. 1719989889



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **PAZMIÑO MOYA VINICIO DAVID**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación “**DISEÑO E IMPLEMENTACION DE UN SISTEMA DE ENTRETENIMIENTO LÚDICO BASADO EN IDENTIFICACIÓN DE MOVIMIENTOS DE MANOS A DISTANCIAS CORTAS USANDO EL SENSOR LEAP MOTION CONTROLLER**” cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 9 de agosto de 2017

Vinicio David Pazmiño Moya

C.C. 1719989889

DEDICATORIA

A Dios, por haber permitido que estuviese en este mundo para disfrutar de la vida y las cosas buenas y malas que ésta tiene, por darme la oportunidad de cumplir un importante logro en mi vida.

A mis padres Marco y Mery quienes me han dado la vida y desde entonces me han sabido guiar y brindado su apoyo incondicional para culminar con éxitos mis propósitos, formándome la persona que hoy en día soy.

A mis abuelitos, los que aun puedo abrazar y a los que sé que desde el cielo me cuidan y protegen, quienes han tenido la fortaleza para sacar a delante a mi familia siendo ejemplo de trabajo y dedicación pese a las circunstancias y adversidades.

A mis hermanos, Jéssica y Anthony quienes han estado a mi lado en las buenas y malas brindándome su apoyo y estar presentes en los mejores momentos de mi vida.

A mi familia, mis tíos y primos quienes se preocupan y siguen cada paso en mi vida, gracias a sus consejos y ejemplo de trabajo para cumplir todas las metas, por los momentos que he comparto con ellos y forman parte importante en mi vida.

A mis amigos, quienes me han acompañado en mi vida y a lo largo de mi carrera.

Vinicio David Pazmiño Moya

AGRADECIMIENTO

Quiero agradecer primero a Dios por permitirme haber llegado y culminado una hermosa etapa de mi vida llena de buenos y malos momentos.

Al apoyo incondicional y el amor que me brindan mis padres Marco y Mery, por darme su guía para crecer como persona y profesional, agradezco su paciencia y comprensión en todo el proceso para cumplir con mi meta.

Agradezco a mis abuelitas Lucrecia y Rosa por su amor y consejo, hermanos Jéssica y Anthony, primos y toda mi familia por formar parte de mi vida y todos los momentos compartidos.

Agradezco a mis compañeros de universidad con quienes compartí gran parte de mi carrera y poco a poco se han convertido en verdaderos amigos de vida, por compartir días de risas y malas noches de trabajos cada semestre, a mi amigo Anthony F. al trabajar como gran equipo en proyectos y conseguir logros y buenas experiencias, a quienes también tuve la oportunidad de conocer y los considero verdaderos amigos Alexander B., Michelle G., Fernanda A. y Leandro A., gracias a ellos por convertir las clases en aulas de diversión y amistad. Al apoyo también, de una gran persona Jessica O. quien supo animarme para culminar el último semestre de inglés.

Finalmente agradezco a la universidad y mis profesores por impartir sus conocimientos para formarme como profesional y a mi tutor de tesis Ingeniero Darwin Alulema por guiarme durante todo el proyecto.

Vinicio David Pazmiño Moya

ÍNDICE DE CONTENIDOS

| | |
|---|-------------|
| CERTIFICACIÓN | ii |
| AUTORÍA DE RESPONSABILIDAD | iii |
| AUTORIZACIÓN | iv |
| DEDICATORIA | v |
| AGRADECIMIENTO | vi |
| ÍNDICE DE CONTENIDOS | vii |
| ÍNDICE DE TABLAS | x |
| ÍNDICE DE FIGURAS | xi |
| RESUMEN | xiii |
| ABSTRACT | xiv |
| CAPÍTULO 1 | 1 |
| 1.INTRODUCCIÓN | 1 |
| 1.1. Antecedentes..... | 1 |
| 1.2. Justificación e importancia..... | 3 |
| 1.3. Alcance del proyecto..... | 4 |
| 1.4. Objetivos..... | 6 |
| 1.4.1. General | 6 |
| 1.4.2. Específicos..... | 6 |
| 1.5. Estado del arte..... | 7 |
| CAPÍTULO 2 | 10 |
| 2.MARCO TEÓRICO | 10 |
| 2.1. Detector de movimiento | 10 |
| 2.1.1. Sensores infrarrojos | 11 |
| 2.1.2. Sensores ultrasónicos | 11 |
| 2.2. Leap Motion Controller..... | 11 |
| 2.2.1. Área de visión y trabajo..... | 12 |
| 2.2.2. Leap Motion sensor..... | 13 |
| 2.3. Unity..... | 16 |
| 2.3.1. Entorno de unity | 16 |
| 2.3.2. Ambiente de desarrollo | 17 |
| 2.4. Ingeniería de software | 18 |

| | |
|---|-----------|
| 2.4.1. Extreme programming (XP)..... | 20 |
| 2.4.2. Modelado del sistema | 21 |
| 2.4.3. Lenguaje unificado de modelado (UML)..... | 22 |
| 2.5. Arquitectura de software | 23 |
| 2.5.1. Arquitectura empresarial | 24 |
| 2.6. Consumir las API de Leap Motion | 26 |
| 2.7. Descripción de Características y Movimientos | 27 |
| CAPITULO 3..... | 29 |
| 3.DISEÑO E IMPLEMENTACIÓN DEL SISTEMA | 29 |
| 3.1. Descripción general del sistema | 29 |
| 3.2. Proceso de desarrollo | 30 |
| 3.2.1. Diagrama de bloques | 30 |
| 3.2.2. Análisis para el diseño del sistema | 30 |
| 3.2.3. Requerimiento del sistema..... | 30 |
| 3.2.4. Requerimientos funcionales | 31 |
| 3.2.5. Requerimientos no funcionales | 32 |
| 3.2.6. Diseño del sistema | 32 |
| 3.3. Proceso de implementación..... | 43 |
| 3.3.1. Escena “Menú Principal” | 43 |
| 3.3.2. Escena “Laberinto” | 45 |
| 3.3.3. Escena “Museo” | 48 |
| CAPITULO 4..... | 53 |
| 4.PRUEBAS Y RESULTADOS | 53 |
| 4.1. Pruebas del sistema | 53 |
| 4.1.1. Evaluación del funcionamiento del sensor | 53 |
| 4.1.2. Evaluación del funcionamiento de la aplicación | 54 |
| 4.2. Pruebas de concepto | 54 |
| 4.2.1. Pruebas de usabilidad..... | 54 |
| 4.3. Análisis de resultados | 55 |
| CAPITULO 5..... | 64 |
| 5.CONCLUSIONES Y RECOMENDACIONES..... | 64 |
| 5.1. Conclusiones | 64 |
| 5.2. Recomendaciones | 66 |
| 5.3. Trabajos Futuros..... | 67 |

REFERENCIAS BIBLIOGRAFICAS 69

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1 Funcionamiento del sensor en distintas condiciones de luz | 15 |
| Tabla 2 Estructura de Elementos | 25 |
| Tabla 3 Estructura de Alocaciones | 25 |
| Tabla 4 Magnitudes en la API Leap Motion | 26 |
| Tabla 5 Muestra utilizada para evaluar el sistema..... | 56 |
| Tabla 6 Número de usuarios evaluando nivel de satisfacción, preguntas 1, 2 y 5 | 60 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1 Dimensiones y ejes coordenados del sensor Leap Motion Controller | 12 |
| Figura 2 Área de visión y área de interacción | 13 |
| Figura 3 Leap Motion Controller, Tools SDK..... | 13 |
| Figura 4 Puntos de interés para el seguimiento posicional | 14 |
| Figura 5 Entorno de desarrollo de Unity..... | 17 |
| Figura 6 Estratos de la ingeniería de Software | 19 |
| Figura 7 Modelo tetraédrico | 22 |
| Figura 8 Arquitectura y diseño de software | 24 |
| Figura 9 Código base para usar API Leap Motion | 27 |
| Figura 10 Ángulos euler o de navegación | 28 |
| Figura 11 Vector de dirección y vector normal | 28 |
| Figura 12 Diagrama de bloques del sistema | 30 |
| Figura 13 Modelo Tetraédrico Modificado | 33 |
| Figura 14 Escenario "Menú Principal" | 34 |
| Figura 15 Diseño gráfico de la escena "Laberinto" | 35 |
| Figura 16 Texturizado del escenario "Museo" | 36 |
| Figura 17 Arquitectura basada en componentes | 38 |
| Figura 18 Diagrama de casos de uso | 39 |
| Figura 19 Diagrama de secuencias del sistema..... | 40 |
| Figura 20 Diagrama de clases de la aplicación..... | 41 |
| Figura 21 Esquema del diagrama de clases por componentes | 42 |
| Figura 22 Fondo de escenario "menú principal" | 43 |
| Figura 23 Interacción entre menús..... | 44 |
| Figura 24 Componentes (script) en el objeto "GestorEscena" | 44 |
| Figura 25 Modelos de juguetes para la escena..... | 45 |
| Figura 26 Modelado del tablero "laberinto" en Blender | 46 |
| Figura 27 Menú del juego "Laberinto" | 46 |
| Figura 28 Elementos que componen el mecanismo del juego | 47 |
| Figura 29 Tablero de laberinto nivel 2..... | 48 |

| | |
|---|----|
| Figura 30 Entorno del escenario "Museo" | 49 |
| Figura 31 Ubicación de objetos para la actividad 1 | 49 |
| Figura 32 Menú de actividades (iglesia Catedral) | 50 |
| Figura 33 Movimientos necesarios para navegar la escena | 50 |
| Figura 34 Menú inicio de la escena "Museo" | 51 |
| Figura 35 Escenas a ser construidas en la aplicación final..... | 52 |
| Figura 36 Funcionamiento en campo de visualización del sensor | 53 |
| Figura 37 Ubicación del <i>Hardware</i> | 54 |
| Figura 38 Ficha de Evaluación del sistema..... | 55 |
| Figura 39 Porcentajes de personas que conocen el sensor | 57 |
| Figura 40 Porcentaje usado de sistema vs Tiempo empleado..... | 57 |
| Figura 41 Resultados de los usuarios sobre la interfaz del sistema..... | 58 |
| Figura 42 Porcentaje de desempeño en dos grupos de evaluación..... | 59 |
| Figura 43 Tabulación sobre la dificultad del uso del sistema | 60 |
| Figura 44 Usuarios evaluando el nivel de satisfacción pregunta 3, 4 y 6 | 61 |
| Figura 45 Tendencia del sistema como herramienta de entretenimiento y aprendizaje | 62 |

RESUMEN

En la actualidad, el auge de los dispositivos tecnológicos se encuentra revolucionando una gran variedad de campos de estudio, por lo cual, la innovación y uso de todos estos avances es necesario en la búsqueda de soluciones que tengan como objetivo brindar y mejorar la calidad de vida de las personas. Actualmente se encuentran un sin número de herramientas didácticas y de aprendizaje implementados mediante sensores gestuales en software 2D, motivo por el cual el presente trabajo busca crear un sistema tanto para el entretenimiento como para el aprendizaje mediante el juego, la aplicación desarrollada es interactiva y con alto apartado visual de entornos en tres dimensiones. La implementación del mismo se realiza en el motor de desarrollo de videojuegos Unity 3D con una programación orientada a objetos y componentes en C# (C Sharp) donde el usuario puede interactuar con entornos virtuales a través de tecnología Interfaz Humano-Computadora (HCI), utilizando el sensor Leap Motion Controller se mapea las manos e identifica sus movimientos y gestos para mostrarlas en pantalla, el software oficial es el Unity Assets para Leap Motion Orión Beta, paquete que con el cual es posible montar escenarios en Unity de forma que es posible manipular objetos e interfaces de usuario.

Palabras Clave:

- **LEAP MOTION CONTROLLER**
- **SENSORES GESTUALES**
- **ENTORNOS VIRTUALES 3D**
- **CONTROLADOR DE GESTOS**
- **MOTOR DE INTERACCION**

ABSTRACT

Nowadays, the boom of technology devices is currently revolutionizing many a wide variety of fields of study, so it is necessary to innovate and use the innovation of all these advances, to find solutions that have as Objective to improve people's quality of life. Nowadays, there are many didactic and learning tools implemented by means of gestural sensors in 2D software, which is why the present work seeks to create a system for both entertainment and learning through play, the application developed is interactive and High-visibility visual environments in three dimensions. The implementation is done in the Unity 3D videogame development engine with C # (C Sharp) object-oriented programming and components where the user can interact with virtual environments through Human-Computer Interface (HCI) technology, using The Leap Motion Controller sensor maps the hands and identifies their movements and gestures to display them on the screen, the official software is the Unity Assets for Leap Motion Orion Beta, a package with which it is possible to mount scenarios in Unity so that it is possible to manipulate objects and user interfaces.

Key Word:

- **LEAP MOTION CONTROLLER**
- **GESTURE SENSORS**
- **VIRTUAL ENVIRONMENT 3D**
- **GESTURE CONTROLLER**
- **INTERACTION ENGINE**

CAPÍTULO 1

1. INTRODUCCIÓN

1.1. Antecedentes

Es un hecho que las Telecomunicaciones han ido tomando fuerza a lo largo del tiempo y es un recurso sumamente invaluable, permitiendo llegar a infinidad de lugares e interconecta millones de personas y mantenerlos comunicados. Esto se debe a los avances tecnológicos en áreas de eléctrica y electrónica los cuales se enfocan a diversos ámbitos de investigación.

Las Tecnologías de la información y la comunicación (TICs) ofrecen una fortaleza con experiencias positivas que va en aumento a nivel mundial. La evolución tecnológica trae consigo dispositivos capaces de explotar los sentidos y percepciones humanas que usados de forma adecuada, brindan un sistema ilimitado de aplicaciones que permiten desarrollar y potenciar una mejor educación. (García Garcés, Navarro Aguirre, López Pérez, & Rodríguez Orizondo, 2014)

Leap Motion Controller es un dispositivo que en 2010 desarrollado un prototipo para identificar el movimiento de las manos e interactuar con dispositivos portátiles (Laazizi Ruiz, 2014), lo cual reemplaza los comandos y dispositivos tradicionales. Se puede interactuar en una superficie cubica semiesférica de 1 metro de radio. Este hardware multiplataforma cuenta con una alta precisión de detección permitiendo un control de hasta 0.01mm de movimiento en los dedos (Fernández, Navarro, & Muñoz, 2013)

Existe una gran variedad de aplicaciones para dar uso al dispositivo Leap Motion Controller, como conocer sus ventajas y desventajas frente a similares sensores de reconocimiento de gestos, mediante la interfaz humano-

computadora (Alfaro Vives & Velarde Robles, 2015) como en el manejo de drones, al igual que en la educación mediante varias aplicaciones para aprendizaje (Laazizi Ruiz, 2014).

El uso de la tecnología para personas con alguna capacidad diferente requiere una usabilidad cualitativa de gran facilidad, hoy en día los dispositivos electrónicos cuentan con una interfaz natural de usuario, como lo es Leap Motion Controller, lo que permite a las personas una accesibilidad más intuitiva y por ende más fácil con el mundo tecnológico (Beltramone, Rivarola, & Quinteros Quintana, 2015).

Existen diferentes tipos de discapacidades sea intelectual, motriz, física o un conjunto de todas estas, Beltramone en su trabajo (2014) lo define como una limitación o restricción para realizar determinada actividad, las interfaces naturales permiten que su uso contribuyan con la movilidad motora del cuerpo y coordinación viso-motriz (2015), resultando de esta forma una fuerte herramienta para la educación y el desarrollo motriz de niños y adultos (Karashanov, Manolova, & Neshov, 2016). Leigh Ellen Potter de la Griffith University en Australia (2013) menciona el uso de Leap Motion Controller para la comunicación de personas especiales a través del lenguaje de señas, obteniendo resultados favorables sin embargo con ciertas limitaciones para la gama completa de signos y señales.

Mientras la tecnología crece aparecen nuevos conceptos y estrategias para mejorar la calidad de vida, así el concepto de *eHealth* (salud en base a tecnología) es cada vez más frecuente al momento de encontrar herramientas y soluciones en la salud mediante la telemedicina como lo menciona José Cepeda (Innovaciones en salud y tecnologías: las cosas claras, 2012). La solución en cuanto a problemática de la motricidad fina en base a un reconocimiento de lenguajes de señas se logra mediante Leap Motion Controller a base de pruebas que muestran la capacidad de seguir el

movimiento de manos y dedos proporcionando una gran precisión (Potter, Araullo, & Carter, 2013).

Hay diversos dispositivos que usan sensores para el reconocimiento y seguimiento de gestos humanos, como el Kinect que se usa en videojuegos (virtualwaregroup, 2013), (Losa, y otros, 2015), (Guna, Jakus, Pogačnik, Tomažič, & Sodnik, 2014) y el control Wii (Kin , Sevcenco, & Yan, 2013); además las librerías abiertas que poseen, los usuarios han desarrollado miles de aplicaciones en diferentes ámbitos como lo plantea Rafael Assis (2012) que menciona en su trabajo, la importancia de la telemedicina en base a las tecnologías e interfaces naturales de usuario como la que provee Kinect para diagnósticos médicos y modelos de virtualización en 3D.

1.2. Justificación e importancia

Se conoce que la tecnología influye enormemente para la sociedad, por lo cual el uso de dispositivos electrónicos para contribuir con herramientas que mejoren la salud y estado de ánimo son imprescindibles, la accesibilidad a estos mecanismos difieren sea por coste, fiabilidad o escalabilidad en el tiempo.

El Ecuador busca una educación de excelencia, por lo que el Buen Vivir exige unas métricas que supere los límites tradicionales de crecimiento integrando; en el Plan Nacional de Desarrollo se plantea seis dimensiones básicas para la planificación, mencionando el acceso universal a bienes superiores el cual establece el derecho a la educación para una mejor calidad de vida (Secretaría Nacional de Planificación y Desarrollo, 2013).

El aprendizaje se da desde el momento que se nace como derecho innato y en el Curriculum de Educación Inicial, el aprendizaje desde niños refleja el correcto desarrollo de sus capacidades, habilidades y destrezas visuales y motoras; debido a esto es importante proveer de herramientas

lúdicas educativas que contribuyan con su crecimiento intelectual (Ministerio de Educación, 2014).

Varios estudios indican que el uso de sensores y dispositivos que poseen su interfaz natural de usuario son una excelente herramienta para entretenimiento, educación, salud, bienestar, etc. Debido a que se ha obtenido grandes resultados en las pruebas realizadas, como por ejemplo usuarios con problemas motores en sus extremidades como se menciona en (Karashanov, Manolova, & Neshov, 2016) donde los autores obtiene conclusiones satisfactorias, al igual que juegos visuales e interactivos para terapias de rehabilitación desarrollados con Leap Motion Controller (Losa, y otros, 2015). Así se logra incrementar la tendencia hacia las interfaces naturales de usuario e impulsar proyectos basado en las nuevas tecnologías como la realidad aumentada (RA) y la realidad virtual (RV) para mejorar las herramientas de aprendizaje mediante avances tecnológicos y lúdicos.

La implementación de aplicaciones y métodos innovadores para la educación mediante el uso de actividades lúdicas y sensores de movimiento como Leap Motion Controller abren un enorme campo de posibilidades para su desarrollo, contribuyendo en el campo de las TICs y bienestar, con beneficios para muchas personas.

El proyecto de investigación, busca desarrollar un sistema de entretenimiento que permita a cualquier tipo de usuarios, realizar actividades recreativas a través de gestos naturales realizados por el movimiento de sus manos, la distancia a la que la persona y el sensor se encuentra son pequeñas lo cual se requiere gran precisión para que la aplicación identifique las manos.

1.3. Alcance del proyecto

El presente trabajo de investigación que se muestra, tiene como principal objetivo el diseño e implementación de un sistema de entretenimiento

lúdico empleando sensor de reconocimiento de gestos Leap Motion Controller para la interpretación e identificación de movimiento de manos y dedos a cortas distancias.

El modelamiento del sistema se plantea realizar un análisis de la ingeniería de software, el cual permite elevar el nivel de abstracción para la construcción de sistemas con procesos autónomos y métodos que se pueden reusar independientemente. Mediante un estudio de las arquitecturas de software se consigue un uso adecuado para estructurar modelos, siendo el elemento principal durante todo el ciclo de un proyecto (Texier, De Giusti, & Gordillo, 2014), (González Huerta, 2014). De esta forma se conceptualiza la necesidad de los modelos para la definición e implementación de las actividades y tareas lúdicas a realizar en el juego.

Mediante el movimiento de las manos se obtendrá las señales utilizando el sensor Leap Motion Controller, compuesto por dos cámaras monocromáticas a 300 fps y tres Leds infrarrojos que permitirán captar los movimientos de extremidades en los tres ejes, cubriendo una volumen de 1 metro radio a la distancia próxima de actuación del sensor, el cual capta todos los movimientos o gestos de la mano con una resolución de hasta 0.01mm teóricamente.

Para una mejor caracterización del dispositivo se utilizara el Kit de Desarrollo de Software (SDK) abierto para Windows a través del cual se permite el procesamiento de las señales provistas por los sensores. Para la creación de la interfaz 3D se pretende emplea Unity, el cual ofrece grandes ventajas como herramienta de desarrollo para la creación de aplicaciones, se puede hacer uso de programación en JavaScript y C# como un lenguaje robusto y un buen modelo de orientación a objetos, tiene el beneficio de ser un software multiplataforma y permite simular eventos físicos reales los cuales contribuirán a que las actividades lúdicas desarrolladas generen una atractiva

herramienta de entretenimiento complementada con aprendizaje de tal manera estas serán mucho más dinámicos y capten la atención del usuario.

Mediante la ingeniería de software para usar modelos establece la estructura del sistema del cual se podrá acceder a las distintas aplicaciones a través de un usuario activo que a su vez, al ser dirigido para el sector de jóvenes-adultos, estos podrán manejar las actividades del juego y explorar las diferentes actividades a realizarse, concluyendo con un análisis de resultado sobre la aplicación desarrollada y el impacto al uso de la tecnología para juegos lúdicos.

1.4. Objetivos

1.4.1. General

- Diseñar e implementar un sistema de entretenimiento lúdico mediante el sensor Leap Motion Controller para que se obtenga una herramienta de recreación identificando movimientos a cortas distancias.

1.4.2. Específicos

- Analizar el estado del arte de los sensores de reconocimiento de gestos.
- Caracterizar el funcionamiento del sensor Leap Motion Controller.
- Diseñar el sistema de identificación de movimientos basados en el Leap Motion Controller y el Kit de Desarrollo.
- Desarrollar el sistema de entretenimiento lúdico usando ingeniería de software y Unity.

- Desarrollar el protocolo de pruebas para la validación del sistema.
- Evaluar los resultados.

1.5. Estado del arte

En (Vega Sosapanta , 2017) se muestra el uso del sensor Leap Motion para la plataforma de Java, se puede conocer el uso de las librerías incorporadas en el software de *Netbeans* las cuales están desarrolladas en un lenguaje de programación distinto al que se usa en Unity 3D, su trabajo muestra el desarrollo de aplicaciones como método auxiliar de enseñanza. Brinda refuerzos de conocimientos para niveles de educación básica, acentuando a la tecnología como herramienta fundamental en el aprendizaje, el trabajo busca un instrumento innovador para impartir conocimientos a través de mandos gestuales realizados con las manos. La propuesta presentada también hace uso del sensor Leap Motion Controller para el aprendizaje pero se diferencia en las aplicaciones de aprendizaje y las interfaces de usuario las cuales se implementan en Unity 3D.

En (Freire Conrado, Pazmiño Moya, De La Cruz, & Tobar, 2015) propone una herramienta lúdica para el desarrollo físico y motor de los niños basado en tecnología con reconocimiento de gestos. Kinect como dispositivo innovador revoluciono la industria de los videojuegos, y mediante dicho sensor se presenta aplicaciones de aprendizaje desarrollados en C# a través de Visual Studio. Se presenta un sistema de aplicaciones para aprender números, colores y figuras representando una herramienta complementaria al aprendizaje tradicional. Las similitudes del trabajo desarrollado es el lenguaje de programación utilizado para generar una herramienta lúdica pero tiene como diferencia una población mayor a la que se encuentra dirigida, ya que está encaminada para el aprendizaje de personas en un rango de edad mayor como jóvenes-adultos.

En (Morales Cruz & Tamayo Sánchez, 2017) se presenta un sistema de enseñanza de lenguaje de señas básico mediante sensores gestuales para que personas con y sin discapacidad auditiva puedan manipular la aplicación, se menciona distintos dispositivos que permiten capturar datos mediante gestos, priorizando al sensor Leap Motion como el adecuado para generar un lenguaje de señas a través de las manos. La aplicación usó librerías de programación en *Python* para la interfaz de usuario en 2 Dimensiones con un atractivo visual enfocado para niños. Las similitudes con la propuesta presentada es el uso de un sensor HCI para capturar el movimiento de las manos y usarlas en aplicaciones de entretenimiento y aprendizaje, pero se diferencia en las librerías utilizadas las cuales permiten crear escenas en un entorno virtual.

En (Yépez Montenegro, 2016) se presenta un sistema de iluminación para una vivienda, manejando un ambiente virtual en el cual a través del sensor Leap Motion se manipula el control de luz en diferentes habitaciones, de esta forma se puede observar como el dispositivo puede comunicar al usuario con entornos de 3D. La herramienta de desarrollo es similar ya que se hace uso de Unity 3D pero la propuesta presentada en el trabajo se diferencia en el tipo de aplicaciones implementadas ya que se enfocan en un sistema de entretenimiento y aprendizaje de los usuarios.

En (Quevedo Pérez, 2016) se incorpora la realidad aumentada y la realidad virtual para realzar tareas de tele-operación remota, haciendo uso de los sentidos para la inmersión en un entorno desarrollado en Unity 3D, de tal forma se incorpora el sensor Leap Motion para mejorar la experiencia de usuario en escenas virtuales. La diferencia con la propuesta presentada es el tipo de escena que se utiliza ya que no se incorpora realidad virtual, sin embargo el proyecto es inmersivo en entornos virtuales ya que hace uso de un escenario el cual puede ser explorado y navegado como juegos de primera persona.

En (Mendoza Ayo, 2016) se implementa un control para un cuadricóptero mediante el dispositivo Kinect, es interesante conocer que las acciones y gestos que se realizan mediante el cuerpo, llámese piernas, brazos, manos, etc., permiten crear aplicaciones para controlar dispositivos a distancia, la investigación desarrollada hace uso de librerías del sensor Kinect para generar comandos los cuales se desarrollaron en el lenguaje de programación C#. A diferencia con la propuesta desarrollada la cual hace uso del sensor Leap Motion Controller para manipular y controlar, dentro de escenas, los movimientos de objetos a través de las manos.

En (Gonzáles Sotomayor, 2014) se muestra el diseño de un juego de entretenimiento implementado en 3D, donde la investigación ayuda al desarrollo del pensamiento de niños entre 7 y 11 años los cuales mediante un juego de laberinto pueden aprender y divertirse. Los juegos lúdicos se pueden desarrollar con distintas tecnologías, siempre que permitan al usuario tener un confort al momento de manipular cualquier aplicación. Las similitudes con la propuesta presentada es la utilización de la herramienta de Unity 3D, sin embargo la diferenciación es que se hace uso del sensor Leap Motion Controller y no de inteligencia artificial para interactuar con las aplicaciones implementadas.

Los estudios analizados previamente brindan un panorama donde la tecnología es una herramienta fundamental inmiscuida en los ámbitos del día a día, el uso de sensores gestuales se puede emplear tanto para crear escenarios virtuales controlados para realizar pruebas de distintas índoles, como para el desarrollo de herramientas de enseñanza y diversión.

CAPÍTULO 2

2. MARCO TEÓRICO

2.1. Detector de movimiento

Un detector de movimiento es un dispositivo electrónico compuesto por diferentes tipos de sensores, los cuales permite identificar la presencia de objetos en movimiento o algún tipo de variaciones en la magnitud de una señal o cambios de fenómenos físicos, estos dispositivos son usados especialmente para la detección de personas dentro del campo de acción del sensor.

Existe un gran campo de aplicaciones para el uso de detectores de movimiento según el sensor que se utilice, en seguridad mediante cámaras de vigilancia y sensores de movimiento ante cualquier tipo de anomalía, entretenimiento a través de sensores infrarrojos identificando gestos o movimientos o, entrando en la región de la domótica para iluminación y comodidad.

Se pueden diferenciar dos clases de sensores, los pasivos que generan señales de salida necesariamente alimentada de una fuente auxiliar o externa para identificar la variación de alguna magnitud o evento físico y los sensores activos, aquellos que necesitan de una señal externa la cual se inyectan de forma autónoma en el medio para identificar algún cambio en dicha señal. En la industria hay una gran cantidad de sensores para medir todo tipo de magnitudes, a continuación se muestran aquellos que se usan para la identificación de movimiento.

2.1.1. Sensores infrarrojos

Las grandes compañías que poseen dispositivos para la detección de siluetas humanas en movimiento usan sensores infrarrojos, dichos sensores mediante el uso de la óptica que poseen los lentes de las cámaras monocromáticas se puede medir la radiación electromagnética emitida por sensores infrarrojos dentro de su campo de visión. Cuando un objeto atraviesa por la luz infrarroja los dispositivos identifican dicha variación y es interpretada para ser enviada como señal eléctrica

2.1.2. Sensores ultrasónicos

Los sensores ultrasónicos usan la emisión de sonido para la identificación y detección de objetos a ciertas distancias, la frecuencia que se usa en este tipo de sensor es mayor a la máxima audible del oído humano debido a la frecuencia de 40KHz en la emisión de la señal. Los pulsos ultrasónicos enviados por el dispositivo rebotan y se reflejan al encontrar un obstáculo por lo cual se procesa y se identifica el tiempo en que la señal se envió y se recibió de tal manera que se conoce la posición de un objeto y si éste varía en el tiempo.

2.2. Leap Motion Controller

Leap Motion Controller es un dispositivo electrónico con una gran sensibilidad para el reconocimiento de gestos en dedos y manos, ya que los identifica con una alta resolución en movimientos muy pequeños, la capacidad del núcleo que lo forma brinda una información tridimensional en un muy amplio campo de visión lo que permite la identificación exacta de las manos. En la figura 1 se muestra como el sensor identifica las coordenadas tridimensionales X, Y, Z. (Nandy, 2016)

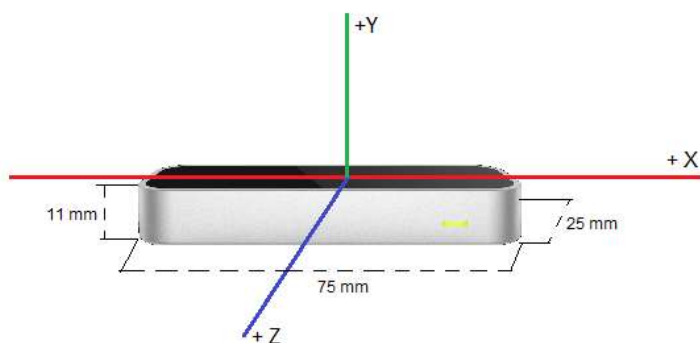


Figura 1 Dimensiones y ejes coordenados del sensor Leap Motion Controller

Leap Motion es un dispositivo con dimensiones muy reducidas, específicamente 75 mm de largo, 25 mm de ancho y 11 mm de altura, siendo, a comparación de otros con interfaz gestual, uno de los más compactos y fácil de utilizar. La forma en el que este dispositivo mapea y captura la ubicación de las manos es mediante luz infrarroja emitida con una longitud de onda de 850 nm por los tres Leds que posee en su hardware. (Belda, Blog de ShowLeap, 2015)

Cuando un objeto se encuentra frente al sensor y es iluminado por inundación de luz infrarroja, se produce una reflexión la cual es capturada mediante las dos cámaras infrarrojas, al formar parte importante en la configuración del dispositivo, estos capturan las imágenes para ser procesadas y reconocer los gestos realizados por las manos y dedos. (Belda & Montalvo Martínez, Blog de ShowLeap, 2015).

2.2.1. Área de visión y trabajo

Para el mapeo correcto sea en 2 o 3 Dimensiones es importante considerar dos aspectos; el primero es conocer la zona de visión del sensor ya que según este se debe asignar las coordenadas correspondientes para que el dispositivo puede captar alguna señal en la zona de trabajo que se conoce como “*interaction Box*”, de esta forma se puede trabajar con cualquier aplicación sin que el sensor pierda de vista los movimientos del usuario (ver figura 2).

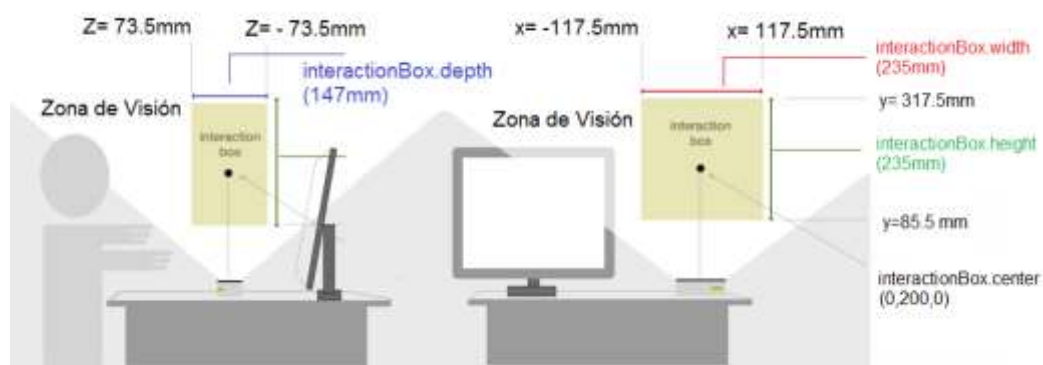


Figura 2 Área de visión y área de interacción

Fuente: (Leap Motion Developer, s.f.)

Cuando se menciona que se debe asignar las coordenadas según la aplicación, se hace referencia a la configuración inicial del centro y el tamaño del área de trabajo con respecto a la zona de visión del sensor ya que de otro modo las esquinas del “*interaction Box*” pueden estar fuera del campo de visión del dispositivo por lo que es muy probable que se pierda el seguimiento de las manos y dedos. (Disqus, s.f.)

2.2.2. Leap Motion sensor

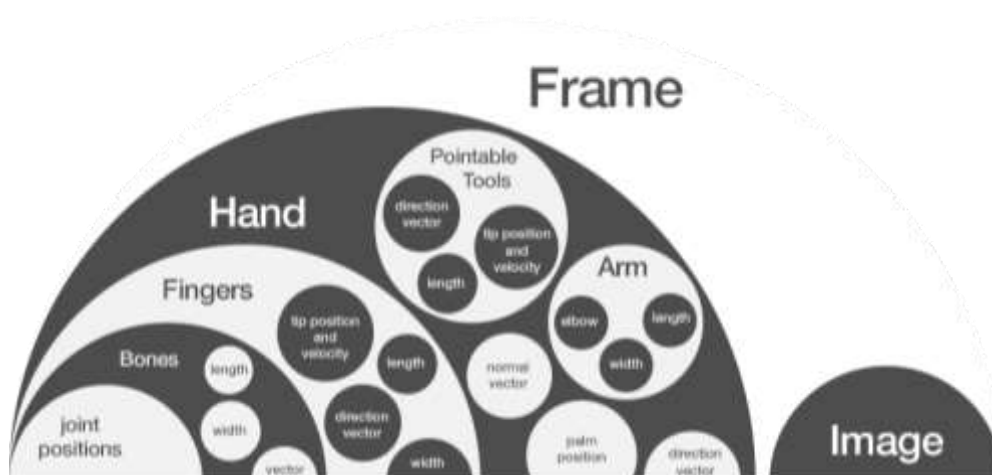


Figura 3 Leap Motion Controller, Tools SDK

Fuente: (Developers, 2016)

El dispositivo cuenta con la interfaz de programación de aplicaciones (API) con subrutinas, funciones y métodos los cuales el usuario puede

acceder para manipular los datos y ser utilizados en cualquier tipo de aplicación, los *frame* o cuadros de imagen son los datos más generales que se puede obtener ya que en este se conoce el seguimiento de entidades como las manos, los dedos y la posición general de los mismos en la escena. (Davis, 2014)

Existen múltiples métodos a los cuales se tiene acceso para hacer lo que se conoce como seguimiento posicional (*positional Tracking*), mediante los modelos matemáticos y algoritmos de seguimiento que posee Leap Motion se puede modelar la posición, características y el movimiento de las manos y dedos para generar los distintos gestos, estos modelos contienen e identifican cada estructura de la mano y lo que la componen como se muestra en la figura 4.

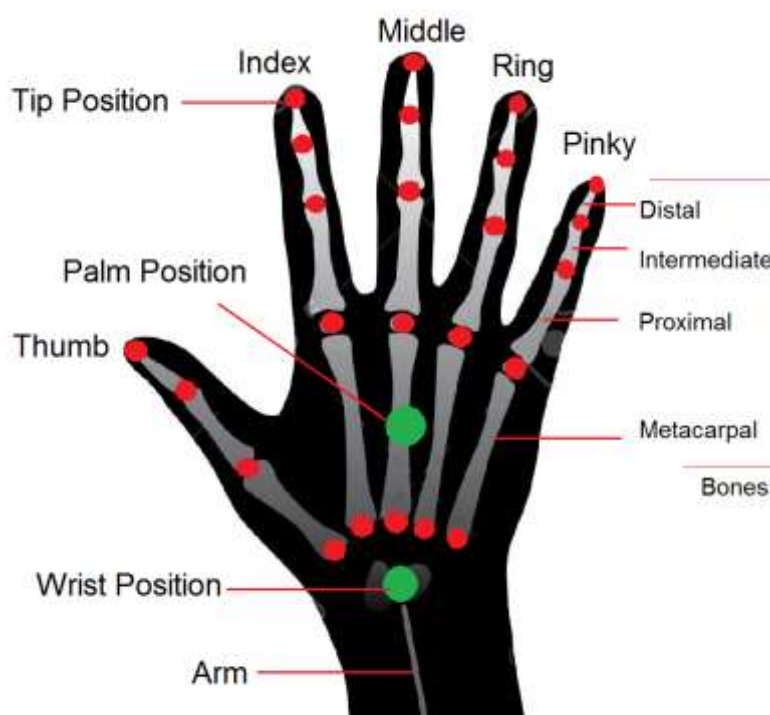


Figura 4 Puntos de interés para el seguimiento posicional

Una vez identificada la mano, cada dedo y las partes que lo componen, el dispositivo, a través de su API, es capaz de reconocer diferentes gestos

predeterminados que se realizan con el giro o movimiento de la mano al igual que los pequeños movimientos de los dedos, esto se hace mediante un proceso de comparación entre el mapeo de cada imagen o *frame* capturado por las cámaras infrarrojas (Nandy, 2016), (Davis, 2014).

La iluminación externa que sufre el sensor es importante para un correcto funcionamiento, al exponer el sensor a distintas luminosidades el desempeño puede fluctuar respecto a los diferentes gestos que reconoce el dispositivo (Morales Cruz & Tamayo Sánchez, 2017).

Tabla 1

Funcionamiento del sensor en distintas condiciones de luz

| Luxes (Luminosidad) | Gestos (%) | | | |
|--------------------------------|-------------------|--------------|---------------|------------------|
| | Señas | <i>Swipe</i> | <i>Keytap</i> | <i>Screentap</i> |
| 6 | 90 | 100 | 60 | 30 |
| 15 | 90 | 100 | 60 | 30 |
| 26 | 90 | 100 | 70 | 30 |
| 80 | 94 | 100 | 80 | 40 |
| 320 | 96 | 100 | 85 | 70 |
| 900 | 100 | 100 | 100 | 100 |
| 1600 | 98 | 100 | 80 | 80 |
| 2400 | 95 | 100 | 60 | 80 |
| 3000 | 94 | 100 | 50 | 80 |

Fuente: (Morales Cruz & Tamayo Sánchez, 2017)

Como muestra la tabla 1 aunque el sensor funcione en condiciones de escasa y extrema luminosidad, es importante acondicionar un buen lugar de trabajar con ambiente controlado de luz natural, de esta forma se podrá aprovechar al máximo la capacidad del dispositivo.

2.3. Unity

Unity Technologies es la empresa que desarrolló un motor gráfico de videojuegos 3D multiplataforma llamada Unity, mediante la cual permite de forma fácil y visual crear un sinnúmero de aplicaciones para desarrolladores independientes. Este software es una poderosa herramienta para crear juegos y simular un mundo cercano a la realidad y de gran calidad, debida a la colaboración entre programadores se puede acceder y hacer uso de una amplia gama de librerías y herramientas para la creación de aplicaciones con texturas, relieves, sistemas, efectos, etc. (Fernández Arroyo, 2013).

Unity es uno de los programas para el desarrollo de videojuegos más versátiles usados hasta el día de hoy, utiliza el motor de gráficos *Direct3D* para Windows y *OpenGL* y *OpenGL ES* para Linux, Mac e iOS, Android respectivamente por lo que se puede encontrar juegos y aplicaciones para todos estos sistemas operativos, Unity también cuenta con una versión de paga y otra gratuita, esta última no cuenta con herramientas más sofisticadas y elaboradas al momento de programar, sin embargo es posible desarrollar aplicaciones de alto grado de calidad (C2Estudio).

2.3.1. Entorno de unity

Unity posee un entorno de desarrollo gráfico, lo que lo hace una herramienta de programación muy intuitiva y fácil de usar, el editor principal de Unity cuenta con una serie de ventanas (ver figura 5), la principal se la conoce como “escena” la que permite observar el avance y desarrollo de la aplicación, las ventanas secundarias por su parte sirven para añadir, cambiar, modificar o categorizar cualquier característica o evento de los objetos y elementos colocados en la escena principal.

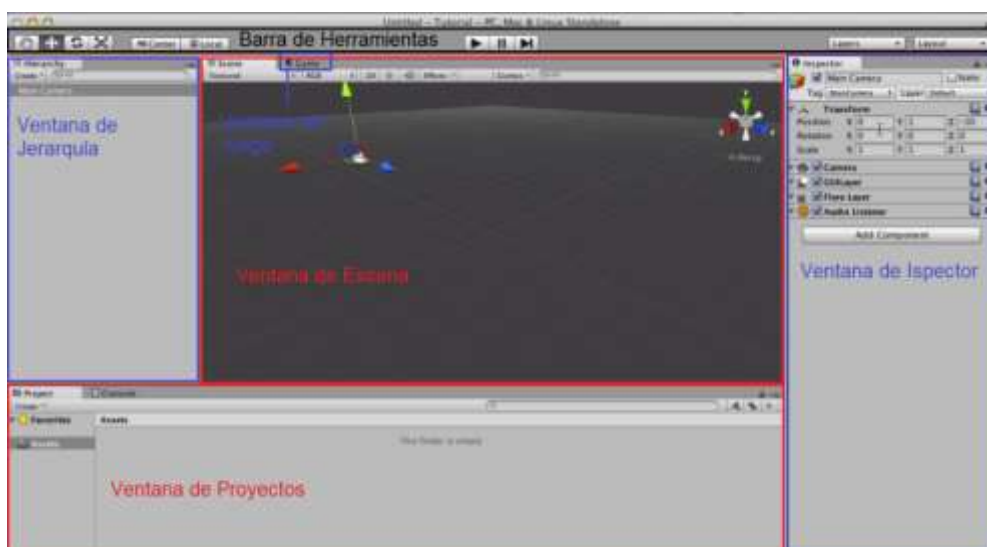


Figura 5 Entorno de desarrollo de Unity

Fuente: (Unity Technologies, 2016)

2.3.2. Ambiente de desarrollo

Trabajar con la herramienta de desarrollo Unity presenta una enorme ventaja para desarrollar aplicaciones ya que al ser multiplataforma, también permite usar diferentes lenguajes de programación orientada a objetos a través de una implementación que está basada en *scripts*, los cuales pueden ser realizados en Java Script, C# (C Sharp), Python y Boo (Alemañ Baeza, 2015).

Por otra parte Unity soporta gran cantidad de paquetes para creaciones 2D y 3D, creaciones de redes y juegos en línea, a su vez permite una compatibilidad con gran variedad de formatos de imágenes, texturas, sonidos, fuentes tipográficas, modelos y animaciones 3D desde programas externos como Blender, Cinema 4D, Autodesk 3D, Studio Max, entre otras. De tal manera que se puede crear aplicaciones con alto profesionalismo para cualquier sistema operativo, es importante destacar que la capacidad de esta herramienta también admite transportarse a otras plataformas sin necesidad de reescribir código nuevo (Fernández Arroyo, 2013).

Otra de las grandes ventajas de trabajar con Unity es la capacidad de realizar cambios en tiempo real, es decir que se puede modificar variables o instanciar objetos al momento de simular la aplicación, es posible reescribir parte de código de los *scripts*, en tiempo de ejecución logrando así corregir o adaptar el comportamiento y desarrollo de la *escena* para posteriormente conocer las modificaciones que se deben realizar.

- **Desarrollador MonoDevelop**

MonoDevelop es el ambiente de desarrollo integrado (IDE) libre y gratuito que viene por defecto instalado en Unity, este IDE combina la operación de un editor de texto con adicionales características para gestionar y depurar las tareas del proyecto que se desarrolla. Este entorno permite la creación de *scripts* en el lenguaje de programación de C# principalmente, el cual es soportado por Unity, gracias a lo cual se ha convertido en un hito para ser un verdadero IDE Multiplataforma (Unity Technologies, 2016).

2.4. Ingeniería de software

A medida que la tecnología avanza, el desarrollo de software actualmente es un tema que se vuelve muy complicado, ya que al implementar sistemas interviene gran cantidad de factores, varios de ellos es la interconectividad a través de internet, la interconexión entre plataformas, la adaptación personalizada del software en diferentes áreas y las necesidades de los diferentes usuarios. Estas necesidades específicas o generales conllevan que diariamente la creación de sistemas o software tenga un alto grado de complejidad para el desarrollador.

Con la tecnología en marcha constantemente los nuevos dispositivos necesitan de un software de soporte que se adapte a cada necesidad, es por eso que existen muchas actualizaciones y versiones en cortos periodos de tiempo, para lo cual la ingeniería del software debe proporcionar múltiples

herramientas y métodos que contribuyan al desarrollo eficaz de un sistema y se adapte rápidamente a cada situación de cualquier producto en cuanto a programación (Cabello Espinosa & Ramos Salavert, 2016).

La ingeniería de software permite definir métodos específicos en semántica y sintaxis para diseñar distintos sistemas los cuales puedan ser compartidos y utilizados en diferentes productos, de tal forma se incorpora el desarrollo de software basado en reutilización que implica “la encapsulación de funcionalidades (previamente desarrolladas y probadas) en elementos que, posteriormente, podrán ser directamente integrados en otros sistemas” y “el desarrollo de estas funcionalidades mediante métodos de ingeniería de software” (Polo Usaola, 2013, pág. 5).

Surge, a partir de la necesidad de implementar sistemas con gran calidad, eficiencia y alto grado de abstracción desde la interfaz hasta su funcionamiento, la integración de procesos, métodos y herramientas para el desarrollo de software (González Sotomayor, 2014), de tal manera que cualquier aplicación pueda ser diseñada, dirigida, construida, operada, administrada y modificada en cualquiera de sus componentes obteniendo un producto de alta calidad.



Figura 6 Estratos de la ingeniería de Software

Fuente: (González Sotomayor, 2014)

Cada modelo trata una parte diferente del sistema y sirven para representarlo en forma general, esto permitiendo generar su propia documentación disminuyendo considerablemente el coste asociado, los

modelos se especifican con un cierto nivel de abstracción e independiente de la tecnología utilizada.

Se requiere contar con lenguajes que permitan representar la estructura y comportamiento del sistema de forma concreta, correcta y precisa al momento de diseñar y desarrollar cualquier aplicación de software, por lo cual es fundamental introducir tres conceptos importantes.

- La **abstracción** permite destacar características importantes y esenciales de un sistema u objeto desde un punto de vista determinado, “ser abstracto no significa en absoluto ser impreciso, el principal propósito de la abstracción es definir un nivel semántico en el que se puede ser totalmente preciso” (Durán Muñoz, Troya Castilla, & Vallecillo Moreno, pág. 10).
- La **expresividad** permite describir términos y elementos de forma clara y sin ninguna ambigüedad, es importante contar con un lenguaje de programación adecuado para el diseño del sistema, descripción de los requerimientos e implementación de la solución. A partir de esto es necesario conocer la **complejidad** del sistema ya que cada herramienta utilizada permitirá que cualquier desarrollador entienda y sea capaz de comprender y razonar sobre el sistema.

2.4.1. Extreme programming (XP)

Es una metodología en el desarrollo de aplicaciones de la ingeniería de software desarrollada por Kent Beck, la cual se centra procesos más ágiles al crear cualquier tipo de software, mediante XP se busca que todos los sistemas implementados sean adaptables a cualquier circunstancia y requisito para que sea sostenible en el tiempo.

Con XP se busca la simplicidad en el desarrollo de código, creando de tal manera diseños sencillos para un desarrollo ágil y facilitar el mantenimiento, esta disciplina también se fundamenta en la comunicación y retroalimentación generando entre desarrolladores un lenguaje más sencillo de entender, es decir crear códigos con baja complejidad para que sean adaptados y mejorados según las necesidades actuales (Villacís, Fuertes, & Santillán , 2016).

En el libro sobre la programación extrema (Beck, 1999) se pueden encontrar los puntos más fundamentales del ciclo de vida de esta herramienta, (1) Fase de Exploración: En la cual se especifican los requerimientos iniciales del sistema para que así se pueda realizar los modelados y la arquitectura del software, (2) Fase de planificación: donde se indican las tareas que se tiene en cada etapa, las versiones que se realizan para llegar al resultado final midiendo el tiempo total en el desarrollo el proyecto, (3) Fase de diseño: En este punto se construye el funcionamiento del programa según los modelos y diagramas de diseño para crear y desarrollar software, (4) Fase de producción: Donde se integra todos los componentes para que funcione el sistema con pruebas unitarias para llegar a un prototipo final y (5) La fase de Mantenimiento: La cual permite adaptar el sistema a los cambios en el tiempo basadas en pruebas de aceptación por clientes y usuarios.

2.4.2. Modelado del sistema

Para el desarrollo de aplicaciones mediante la ingeniería de software es necesario construir un modelo que rija el comportamiento total del sistema, para el cual se estudia el modelo tetraédrico que se basa las acciones más fundamentales para crear un programa (Villacís, Fuertes, & Santillán , 2016).

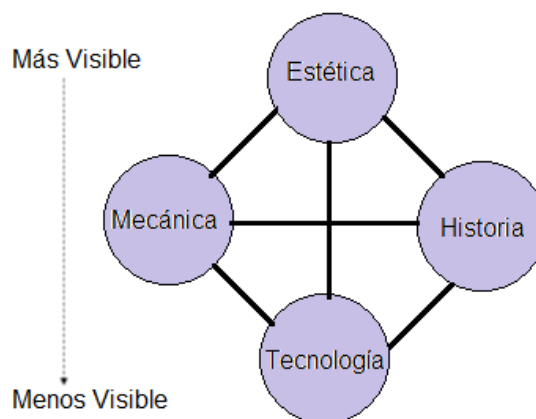


Figura 7 Modelo tetraédrico

Fuente: (Villacís, Fuertes, & Santillán , 2016)

Mediante el modelo de la figura 7 se determina los distintos elementos que permiten la interacción entre el usuario y la aplicación de tal manera que se puede evaluar, a través de cuatro secciones importantes la calidad del sistema. (1) Mecánica: Describe las reglas del juego e indica las acciones necesarias para que el jugador manipule toda la interfaz de la aplicación, a través de esta etapa se materializa el objetivo a cumplir el usuario dentro de la aplicación. (2) Estética: Muestra la sensación de confort del usuario dentro del juego, a través de los sentidos se llega a identificar se la percepción del jugador es buena o mala para hacer los cambios y modificaciones necesarias. (3) Tecnología: Se basa específicamente en todo el hardware y software que requiere el sistema de tal manera que funcione correctamente en cualquier escenario, este punto del modelo presenta los niveles y etapas que constituye la aplicación. (4) Historia: Analiza los eventos que suceden en el desarrollo de la aplicación, describe la narrativa para que el juego finalice en todas sus etapas.

2.4.3. Lenguaje unificado de modelado (UML)

Es un estándar abierto y el más utilizado de la industria para modelar software. Aunque UML sea flexible y permita diseñar sistemas de forma gráfica, en ciertas ocasiones se requiere un lenguaje de uso más específico

representar los conceptos de ciertos dominios particulares, este lenguaje de modelado permite mejorar los procesos de construcción de cualquier sistema (Pons, Giandini, & Pérez, 2010) las ventajas que se obtiene son amplias:

- **Incremento en la productividad**, genera menor coste en el desarrollo de software ya que antes de implementar se documenta de forma visual para detectar cualquier error antes de desarrollarlo. Es posible adaptar al nuevo sistema se reduce el tiempo de implementación y se mejora la productividad del desarrollador.
- **Adaptación a los cambios tecnológicos**, el creciente progreso tecnológico conlleva a que el soporte de software se vuelva obsoleto con gran rapidez, por lo que un correcto uso de los modelos UML soluciona los problemas a través de su arquitectura de modelado para mantenerse dispuesto a cambios en su implementación de forma oportuna, al igual que se vuelve flexible para presentar diferentes ideas en la aplicación final.
- **Mejoras en la comunicación**, en cuanto a usuarios y desarrolladores ya que los modelos muestran un detallado mapa del sistema, facilitando de tal forma el entendimiento total de cualquier aplicación, por lo cual da origen a mayores discusiones productivas que permitan el mejoramiento en los diseños del sistema.

2.5. Arquitectura de software

La arquitectura de software es fundamental para diseñar cualquier sistema ya que es la base en la cual se construye las aplicaciones, se toma en cuenta los aspectos más importantes para el funcionamiento del programa y luego se completa aspectos generales, de esta manera se puede establecer los niveles, estrategias y herramientas necesarias para implementar un sistema que cumpla con los requerimientos funcionales y no funcionales.

Permite establecer los dispositivos de hardware necesarios al igual que las interfaces de software que manejarán los usuarios.

Una arquitectura representa decisiones significativas al momento de elaborar un sistema de software, dichas decisiones componen la estructura del sistema y suelen ser difíciles de modificar ya que se las debe tomar de forma temprana, mediante la arquitectura de software los diseños que se realicen muestran un nivel estratégico alto que definen los principios que guían el desarrollo del sistema. Para entender la diferencia entre la arquitectura y el diseño (ver figura 8) los diagramas UML muestran claramente cuál es el alcance de cada uno (Escobar, Brey, & Martins, 2017).

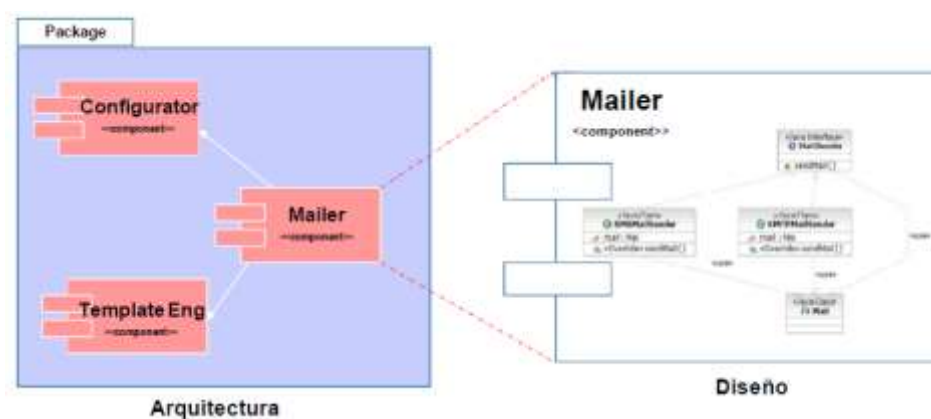


Figura 8 Arquitectura y diseño de software

Fuente: (Escobar, Brey, & Martins, 2017)

Una arquitectura lleva a estructurar y organizar los requerimientos del cliente definiendo componentes que en el proceso de diseño se lleva a cabo para proceder a la implementación, todo un sistema se puede desglosar en múltiples partes para generar sus distintos diagramas llevando a mostrar todos los modelados la misma arquitectura del sistema.

2.5.1. Arquitectura empresarial

Es la primera decisión en ser tomada y es en la cual se gestiona de forma correcta la conexión de los distintos elementos que forman el sistema,

dentro de esta es necesario realizar una arquitectura de infraestructura la cual permite y se relaciona con el uso de hardware y software, de tal manera se emprende con una arquitectura de aplicación donde se especifica más a fondo el desarrollo de un software en particular.

- **Proceso de arquitectura**

Para un proceso arquitectónico de software existen un conjunto de actividades que serán ejecutadas en la mayoría de casos, aunque para cada proyecto existirá su particularidad. La forma en que se organiza describe dos estructuras fundamentales como se puede diferenciar en las tablas 2 y 3.

Tabla 2

Estructura de elementos

| | |
|--------------------|---|
| Módulos | Unidades conceptuales de primer orden que guían el desarrollo |
| Componentes | Bloques técnicos que permitirán llevar a cabo la funcionalidad de los módulos |
| Conectores | La manera en que permiten conectar los módulos y componentes |

Fuente: (Escobar, Brey, & Martins, 2017)

Tabla 3

Estructura de alocaiones

| | |
|-----------------------|---|
| Despliegue | Alocaciones de los módulos, componentes y conectores; Compilación, empaquetado. |
| Ejecución | Alocaciones entre las unidades de despliegue. |
| Implementación | Define las unidades de desarrollo y cómo van a ser construidas |

Fuente: (Escobar, Brey, & Martins, 2017)

2.6. Consumir las API de Leap Motion

Una vez importado los paquetes de Leap Motion a un proyecto de Unity se puede acceder a las librerías del SDK del sensor, la clase principal de la cual se obtiene toda la información de las manos y dedos (posición, velocidad, orientación, etc.) es “*Frame*”, ya que de cada fotograma que se captura se obtiene datos recogido del sensor. Para instanciar cualquier un *frame* primero de se debe crear un objeto de tipo *Controller* del cual se puede verificar que el dispositivo esté conectado para generar los fotogramas. Se debe considerar que la *Application Programming Interface* (API) de Leap Motion mide las magnitudes físicas con las unidades que se muestra en la tabla 4 (Leap Motion Developer, s.f.).

Tabla 4
Magnitudes en la API Leap Motion

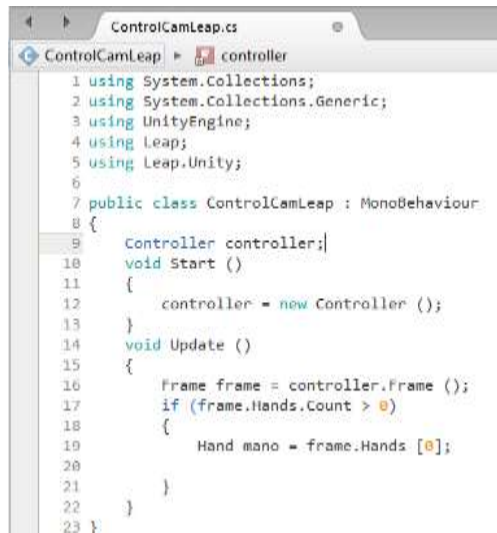
| | |
|------------------|--|
| Distancia | milímetros |
| Hora | microsegundos (a menos que se indique lo contrario) |
| Velocidad | milímetros/segundo |
| Ángulo | radianes |

Fuente: (Developers, 2016)

Después de instanciar un objeto *frame* de la clase *Frame* se puede acceder a la información de varios objetos básicos registrados en el sistema Leap Motion como son la de las clases *Hand*, *Arm*, *Bone*, *Finger* (datos de las manos, brazo, hueso, dedos). Al usar Unity y su desarrollador MonoDevelop el cual se programa con código C# implementados en scripts se puede hacer uso del método *Update ()* que captura automáticamente datos de cada *frame* de la aplicación.

En la figura 9 a continuación se muestra un ejemplo de código con el cual se puede empezar a hacer uso de los datos que proporciona Leap Motion y los gestos que podemos generar en todas las aplicaciones, es importante

denotar que como en cualquier código se debe llamar a las librerías, en este caso, con la palabra *using*: ***using Leap; using Leap.Unity;***



```

ControlCamLeap.cs
ControlCamLeap > controller
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using Leap;
5 using Leap.Unity;
6
7 public class ControlCamLeap : MonoBehaviour
8 {
9     Controller controller;
10    void Start ()
11    {
12        controller = new Controller ();
13    }
14    void Update ()
15    {
16        Frame frame = controller.Frame ();
17        if (frame.Hands.Count > 0)
18        {
19            Hand mano = frame.Hands [0];
20        }
21    }
22 }
23 }

```

Figura 9 Código base para usar API Leap Motion

2.7. Descripción de Características y Movimientos

Existe una gran cantidad de clases en Leap Motion de las cuales se puede obtener diferentes características y atributos de los datos que se leen mediante el sensor, como por ejemplo si la mano usada es derecha o izquierda, en qué posición se encuentra, el dedo usado es el anular de la mano izquierda, etc., y un sin número de opciones más. Para el proyecto desarrollado se describe tres movimientos de la muñeca importantes que fueron implementados; estos son *Pitch*, *Roll* y *Yaw*, los cuales son un tipo de ángulos de Euler que sirven para la navegación y describen la orientación de un objeto en tres dimensiones.

A través de estos ángulos de rotación que se puede obtener de las manos del jugador se realiza acciones como FPS para navegar dentro de la escena “Museo” y la de controlar el movimiento del tablero dentro de la escena “Laberinto”, en la figura 10 se puede observar que tipo de movimientos y respecto a que eje es cada uno.

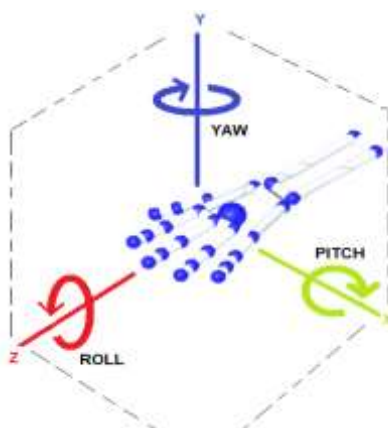


Figura 10 Ángulos euler o de navegación

Estos ángulos proveen la orientación de la mano a través del objeto “mano” de la clase *Hand*, los valores se obtiene mediante las direcciones de vectores normales que el SDK calcula al capturar la palma de la mano. La clase *Vector* define estas funciones de la siguiente manera:

- `float pitch = hand.Direction.Pitch;`
- `float yaw = hand.Direction.Yaw;`
- `float roll = hand.PalmNormal.Roll;`

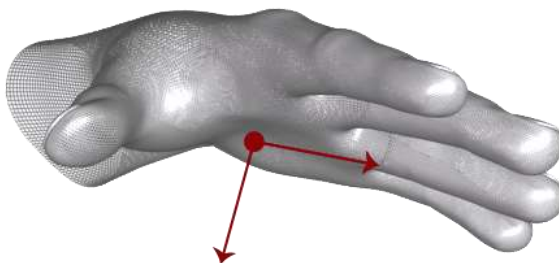


Figura 11 Vector de dirección y vector normal

Fuente: (Developer, s.f.)

CAPITULO 3

3. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

3.1. Descripción general del sistema

El sistema de entretenimiento lúdico implementado en el motor de desarrollo de videojuegos “Unity” tiene como objetivo el uso del sensor Leap Motion Controller y las características que le permiten al usuario interactuar de forma física con el sensor para adentrarse en el mundo de los juegos, las librerías y módulos del hardware Leap Motion permiten manipular y crear un entorno virtual y distintos escenarios en el cual es posible identificar y procesar el movimiento de las manos a corta distancia, concluyendo con el desarrollo de una aplicación donde se puede interactuar con el mundo 3D a través de los gestos realizados con las manos del usuario.

Para el sistema se realizan tres escenarios de los cuales dos de ellos serán las aplicaciones; el primer escenario es la presentación principal de la interfaz la cual despliega un menú para realizar configuración básicas y acceso a las dos aplicaciones, el segundo escenario es un juego llamado “Laberinto” donde el usuario a través de los movimientos y rotación de la muñeca de la mano podrá guiar un tablero en forma de laberinto, por último el tercer escenario será una aplicación llamada “Museo” en la cual el usuario podría navegar como juegos en primera persona por una habitación ambientada en forma de museo de las iglesias de Quito.

3.2. Proceso de desarrollo

3.2.1. Diagrama de bloques



Figura 12 Diagrama de bloques del sistema

3.2.2. Análisis para el diseño del sistema

Es importante analizar los componentes necesarios a utilizar en el desarrollo del sistema, el diseño de la interfaz que el usuario manipulará y los elementos esenciales para la implementación de la aplicación, tomando en cuenta la simplicidad que requiere para que el funcionamiento básico sea intuitivo y conociendo las características del sensor Leap Motion Controller para realizar gestos.

3.2.3. Requerimiento del sistema

Los elementos necesarios que se deben incorporar tanto en hardware como en software para que el sistema funcione de forma adecuada son los siguientes:

Dispositivo Leap Motion Controller.- Sensor que permite obtener los datos del entorno y trabaja con gran robustez en cualquier luz ambiental, realiza una captura de imágenes a través de sus cámaras sensibles a luz infrarroja y realiza el seguimiento de las manos en distancias cortas.

Computador.- Equipo que permite realizar el procesamiento adecuado de la información proporcionada por el sensor para el flujo adecuado del sistema,

es necesario un sistema operativo con Windows, procesador Core i3 o superior y un puerto USB 2.0. **Leap Motion Software.**- Controlador que permite la instalación del hardware para que el computador lo reconozca. **Unity (Software).**- Motor de Videojuegos multiplataforma que permite desarrollar aplicaciones graficas en 2D y 3D. **Leap Motion SDK.**- Kit de desarrollo de Software que contiene las licencias y librerías necesarias para ejecutar ambientes virtuales de Leap Motion sobre la plataforma de Unity. **Módulos Unity.**- Paquetes desarrollados con librerías de Leap Motion para ser implementados en Unity compatibles con PC para escritorio, Gafas de realidad virtual y realidad aumentada. Módulos utilizados: *Interaction Engine* y *UI Input*.

3.2.4. Requerimientos funcionales

Estos requerimientos definen las acciones y funciones necesarias que el usuario realizará en todas las etapas de la aplicación a medida que se vaya ejecutando los escenarios, a continuación se presenta las opciones disponibles que se incorporaron debido a la necesidad que los juegos necesitan dentro de su interfaz gráfica de usuario.

1. **Laberinto:** Permite ejecutar la primera escena de la aplicación, la cual muestra el primer nivel del juego “Laberinto”.
 - **Opción Continuar:** Permite dar paso a la realización del juego
 - **Opción Repetir:** Inicializa todas las variables y ajustes iniciales para repetir la partida actual.
 - **Opción Menú Principal:** Permite volver al escenario inicial donde se encuentra las opciones de menú principal.
2. **Información:** Permite ejecutar otra ventana de interfaz de usuario la cual contiene información sobre las instrucciones de sus respectivos juegos “Laberinto” y “Museo”.

3. **Museo:** Ejecuta la escena del juego llamado “Museo” donde se puede explorar un ambiente virtual.
 - **Opción “Iglesia” (nombre de cada iglesia):** Permite encontrar las actividades necesarias para la interacción con el juego y el ambiente 3D, se puede acceder a todas las iglesias que se encuentran en el escenario.
4. **Opción Botón “Escape”:** Esta función ejecuta un menú de pausa para realizar cualquier opción que permite la escena.
5. **Ajustes:** Mediante el cual el usuario puede elegir una resolución de pantalla enlistada para modificar el tamaño de visualización de la ventana de la aplicación
6. **Salir:** Permite que el usuario pueda salir del juego y la aplicación se cierre, esta opción se puede encontrar en todos los escenarios del sistema.

3.2.5. Requerimientos no funcionales

El sistema posee distintos escenarios en los cuales el sensor funcione con fluidez para que el funcionamiento sea óptimo para el usuario, cada aplicación poseerá sus reglas las cuales se ejecuten por acciones directas del jugador. Mediante el sensor conectado al computador el usuario podrá hacer uso de una mano o las dos al mismo tiempo según las instrucciones de cada juego. El sistema contiene acciones de FPS (*First Person Shooter*) al igual que acciones en tercera persona según requiera la implementación de cada aplicación.

3.2.6. Diseño del sistema

El diseño del sistema detalla la estructura de todo el sistema, al igual que los diagramas de su funcionamiento, se especifica y se define el comportamiento completo de la aplicación, las acciones y las reglas

requeridas por el usuario para que navegue a través de todas las escenas del juego.

3.2.6.1. Modelo tetraédrico modificado

Los modelos permiten crear un esquema en donde el sistema funcione acorde a las necesidades planteadas, para el caso de la aplicación desarrollada se hace uso del modelo tetraédrico modificado (Villacís, Fuertes, & Santillán , 2016) en el cual se ubican parámetros adicionales a las cuatro etapas principales del modelo mencionado como se puede ver en la figura 13.

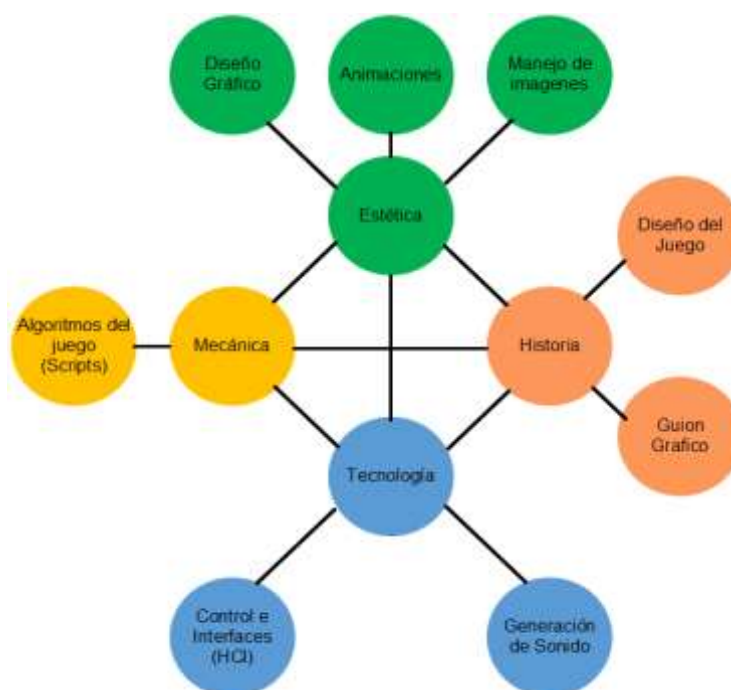


Figura 13 Modelo Tetraédrico Modificado

Fuente: (Villacís, Fuertes, & Santillán , 2016)

- **Diseño del juego**

La historia del juego permite conocer todos lo relacionado con la versión final de la aplicación, para lo cual el diseño del juego estructura todas las escenas que se implementarán al igual que se definen los niveles que existen, a partir de esto se puede generar más escenarios con la misma estructura si así se requiere. El

sistema de entretenimiento cuenta con tres escenarios, “Menú Principal” donde se realiza algunas configuraciones y se conoce las instrucciones para navegar los juegos y dos aplicaciones “Laberinto” y “Museo” las cuales sirven de entretenimiento y aprendizaje respectivamente para la persona que escoja estos escenarios.

- **Guion gráfico**

Esta etapa permite modelar las opciones graficas que permite el funcionamiento del juego, es un esbozo de los escenarios que contendrá, las opciones que son necesarias para que el usuario modifique ciertos parámetros de las aplicaciones y la apariencia que puede tener los menús para el flujo general del juego, como se ilustra en la figura 14.



Figura 14 Escenario "Menú Principal"

- **Diseño gráfico**

En el diseño gráfico se incorpora la parte más importante para que la aplicación atrape al usuario, es decir que mediante el correcto uso de ilustraciones, formas, modelos 3D, luces, texturas, entre otras, el entorno desarrollado llega a tener éxito. De esta forma se debe hacer un diseño tanto atractivo visualmente como intuitivo en su funcionamiento. En la figura 11 se puede observar uno de los escenarios para implementarse. Es importante

considerar las ubicaciones y tamaños correctos en la interfaz de usuario para que la manipulación sea sencilla y rápida, de igual forma el espacio 3D debe contener buena iluminación y los elementos necesarios para no llegar a saturar una escena.



Figura 15 Diseño gráfico de la escena "Laberinto"

- **Animaciones**

Al desarrollar el sistema de entretenimiento en Unity, este motor de videojuegos 3D permite incorporar un sin número de animaciones para cada objeto que sea puesto en la escena, desde los personajes que se incorporen en un videojuego hasta detalles de animación en letras, botones y cualquier objeto de la interfaz de usuario, para el proyecto se incorporó animaciones al accionar un botón del GUI y en los cambios de menú para que de esta forma sea más interactiva con el usuario.

- **Manejo de imágenes**

No solo hace referencia a la incorporación de imágenes en 2D, al igual que en juegos en tres dimensiones el uso de texturas permite crear escenas más fieles a la realidad, por lo cual en las distintas aplicaciones se ha adecuado de forma correspondiente a cada ambiente: para el "Laberinto" por ejemplo se texturizo con colores y formas más cercanas a un cuarto de juegos, mientras que

en “Museo” las ilustraciones incorporadas presentan una iluminación tenue con apariencia más realista a un lugar de visita turística y de conocimiento (ver figura 16).

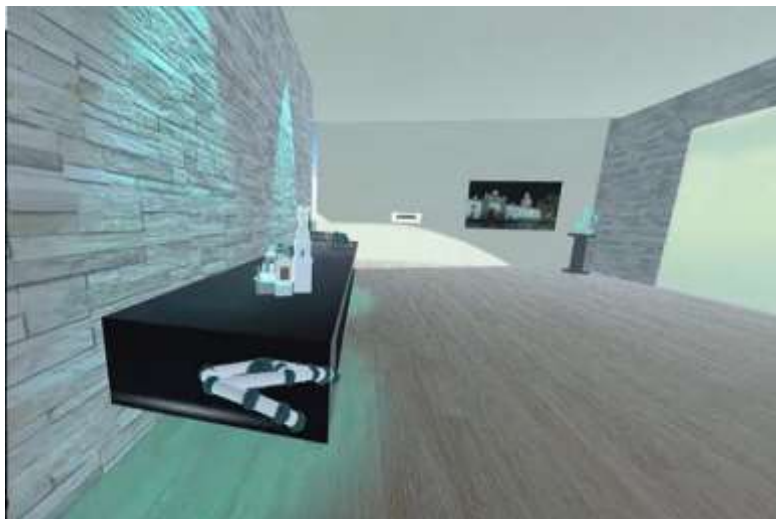


Figura 16 Texturizado del escenario "Museo"

- **Generación de sonido**

Es importante que cualquier aplicación contenga archivos multimedia, para el desarrollo de este sistema se ha incorporado distintos archivos de sonido que permiten dar más realismo a las aplicaciones; por ejemplo el dar *click* en un botón se podrá escuchar tal cual se conoce en la vida real, al iniciar los escenarios del mismo modo se puede percibir música de fondo como en los juegos que existen actualmente. Para complementar la interacción de los usuarios, otros de los archivos multimedia son videos informativos y de aprendizaje sobre la cultura e historia de las iglesias ubicadas en la escena “Museo”.

- **Control de interfaces**

Cualquier aplicación desarrollada necesita incorporar algún tipo de interfaz para que el usuario manipule e interactúe con la aplicación. Para el sistema implementado el control de interfaz

utilizado es *Human Computer Interaction* (HCI), es decir el sensor Leap Motion Controller. Los HCI permiten el intercambio de información mediante software entre el usuario y la computadora, esta disciplina se encarga de analizar e implementar los componentes tecnológicos interactivos que permita una usabilidad sencilla de usar y fácil de recordar, por lo cual, el sensor utilizado es el más óptimo para ser incorporado en este sistema.

- **Algoritmos del juego (Scripts)**

Los algoritmos permiten desarrollar una secuencia de acciones que la computadora debe realizar para llegar a un fin, estos permiten la funcionalidad del juego para que el usuario tenga una experiencia real al interactuar con la computadora, estos códigos desarrollados dentro de Unity 3D se los conocen como *script* y son componentes que forman parte de un *GameObject* (objeto de la escena) para que cada objetos tenga un comportamiento general o particular dentro de la escena, es decir que los códigos o *script* son los algoritmos que rigen el funcionamiento del sistema.

Dentro de la aplicación existen varios *scripts* que permiten la jugabilidad del proyecto; por ejemplo los que permiten cambiar de escenarios, dar acciones a cada botón del GUI, contabilizar las vidas dentro del juego “Laberinto”, acceder a las librerías del sensor Leap Motion Controller para usar los gestos de las manos en una escena, realizar acciones de FPS, etc. Como se especifica en el capítulo 2, se puede generar código en lenguaje de Java Script o C#. Para este proyecto se hace uso de este último, debido a que es la misma sintaxis utilizada en Visual Studio.

3.2.6.2. Modelo arquitectónico

Unity permite crear una estructura para generar software de manera organizada y ordenada, su arquitectura basada en componentes permite un diseño y desarrollo de cualquier sistema con alto nivel de abstracción. Se enfoca en componentes funcionales o lógicos donde se expone interfaces bien definidas. La arquitectura en la que se desarrolla el sistema se puede observar en la figura 17, permitiendo implementar aplicaciones modulares y extensibles.

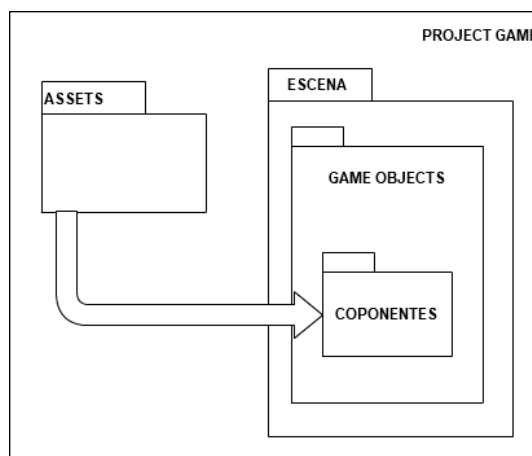


Figura 17 Arquitectura basada en componentes

- *Assets*. - Archivo creado fuera o dentro de Unity que se lo puede incorporar al proyecto en desarrollo.
- *Escena*. - Es el ambiente principal de un proyecto en Unity, la cual contiene todos los objetos del juego.
- *Game Object*. - Son los objetos fundamentales en Unity y son contenedores para almacenar componentes (caracterizan objetos).

3.2.6.3. Modelos UML

Los modelos UML es el lenguaje estándar que representa la notación de sistemas orientados a objetos o basados en componentes. Este lenguaje define claramente la composición de la arquitectura de un sistema permitiendo entender mediante diagramas la estructura de cualquier aplicación de

software. La versión de UML 2.0 permite representar trece diagramas distintos que modelan el problema y la solución, a continuación se presentan los más importantes para diseñar el presente sistema.

Diagrama de casos de uso

Los diagramas de casos de uso muestran una descripción de las actividades y procedimientos para el uso de la aplicación, de forma sencilla los usuarios comprenden de forma detallada las instrucciones que se pueden realizar en el sistema. El diagrama de la figura 18 muestra la interacción que el usuario puede realizar sobre la aplicación del sistema propuesto.

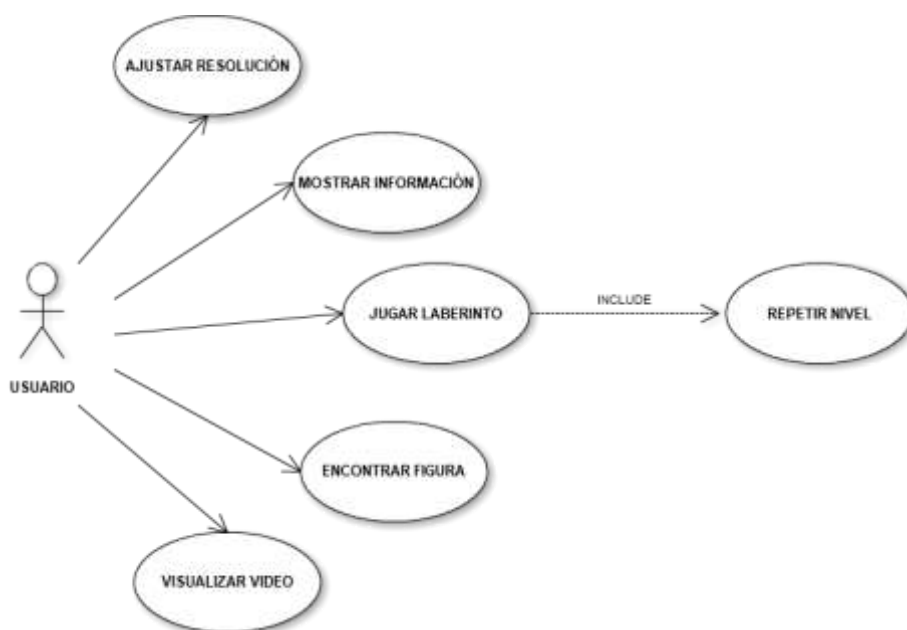


Figura 18 Diagrama de casos de uso

Diagrama de secuencias

Basada en la arquitectura de Unity los diagramas de secuencias desarrollados muestran la interacción en el tiempo entre los componentes y los objetos puestos en las escenas. El diagrama a continuación presente las acciones que transcurren secuencialmente dentro del sistema (ver figura 19).

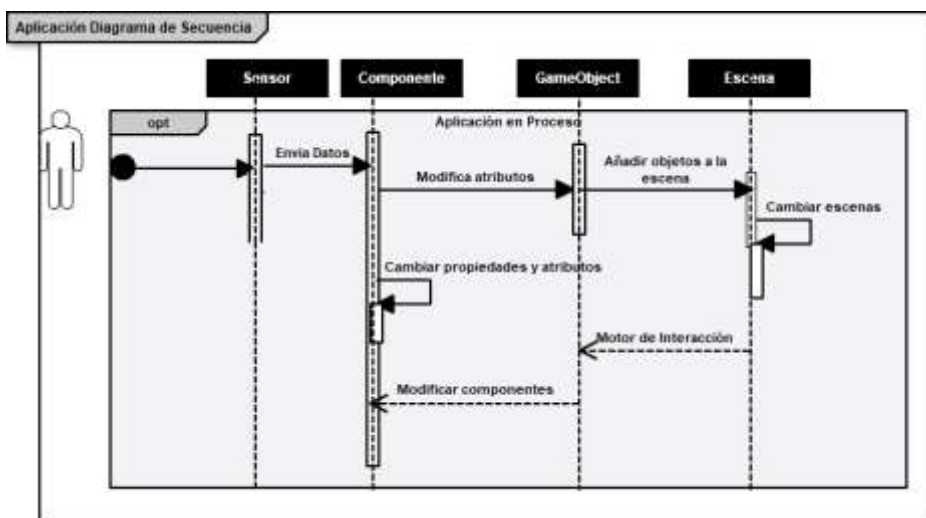


Figura 19 Diagrama de secuencias del sistema

Diagramas de clases

Los *scripts* desarrollados para el sistema permiten definir una clase de objeto, es decir, como Unity posee una arquitectura basada en componentes, éste se crea cuando el *script* es arrastrado al *GameObject* el cual se encuentra en la escena, dicho componente es una instancia de la clase que contiene el *script*, por consiguiente se puede crear tantas copias como se desee manteniendo sus propiedades y métodos definidos en el código, y cada objeto tendrá un valor diferente almacenada en esa propiedad. Cada clase implementada hereda de *MonoBehaviour* (clase principal de Unity) por lo tanto se puede acceder a otros objetos para manipular su comportamiento de forma que el sistema funcione correctamente.

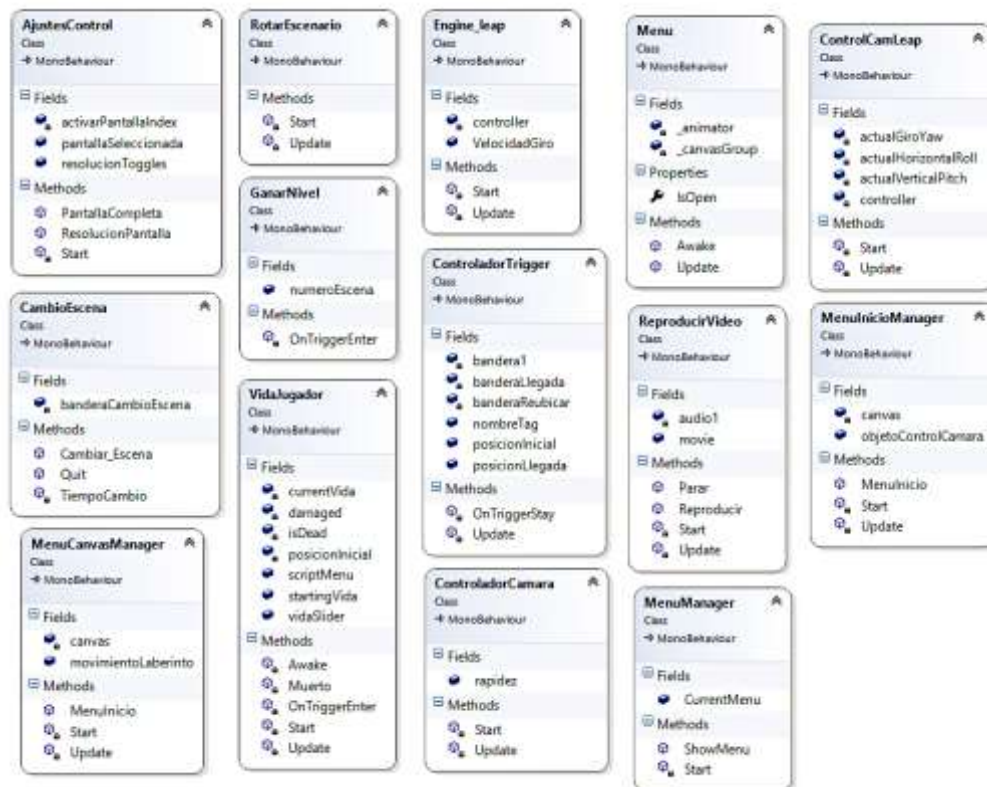


Figura 20 Diagrama de clases de la aplicación

Las clases (ver figura 20) utilizadas permiten manipular los objetos de cada escena, validando la conducta propia del sistema, como se menciona anteriormente cada clase, al incorporarse dentro de un objeto (*GameObject*) entra a formar parte de sus componentes caracterizándolos y diferenciándolos unos de otros. Al poder reutilizar las clases como componentes, algunas se encuentran en más de dos escenas del sistema.

A continuación se presenta un esquema de los diagramas de clases y su distribución dentro de todas las escenas de la aplicación, estas clases se representan como componentes los cuales permiten instanciar un tipo de objeto para cada parte del sistema (ver figura 21).

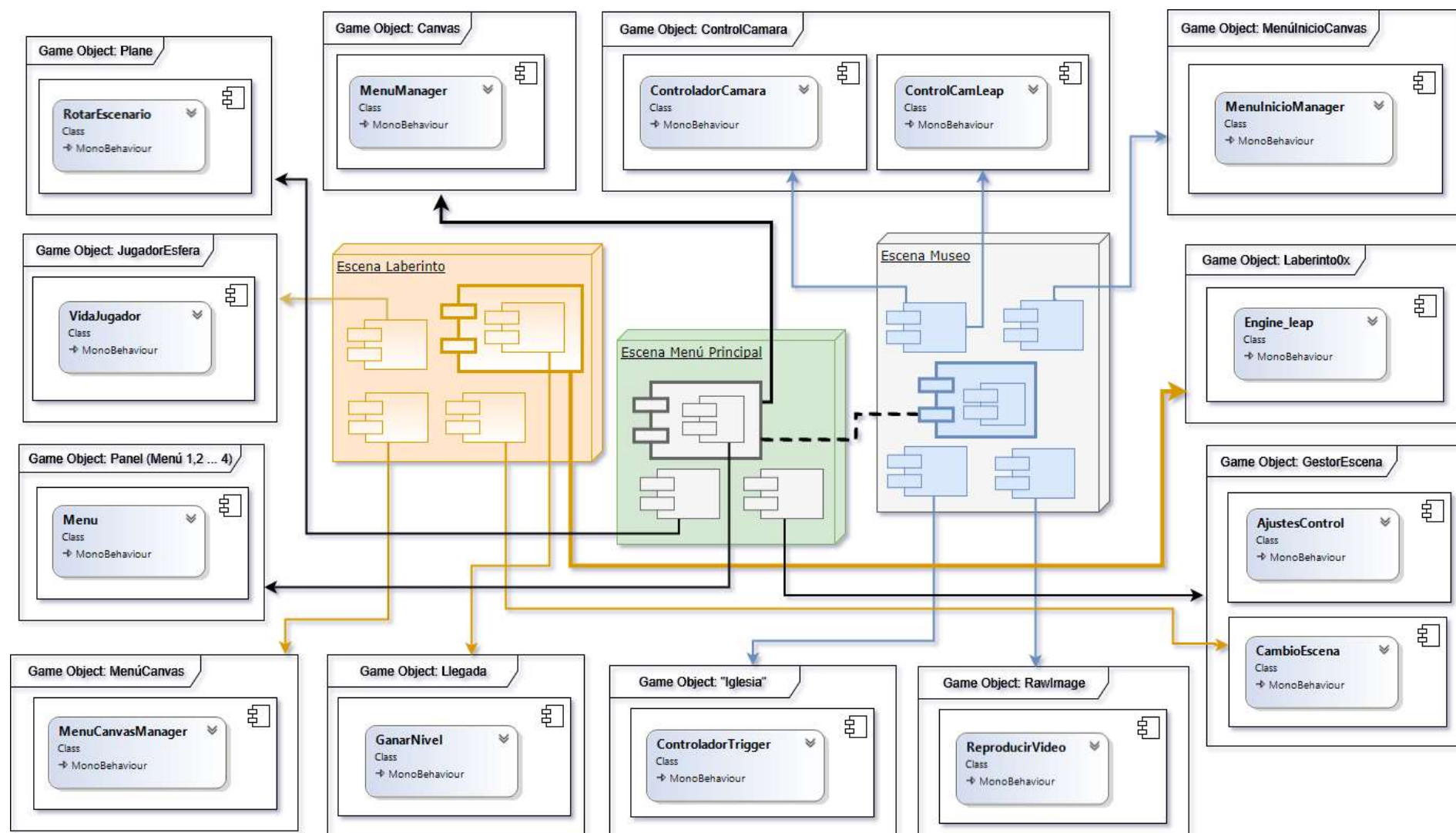


Figura 21 Esquema del diagrama de clases por componentes

3.3. Proceso de implementación

3.3.1. Escena “Menú Principal”

En la escena llamada menú principal se puede encontrar un fondo virtual que consta de una superficie rodeada de objetos como especie de paredes iluminadas, se puede encontrar dos cuadros y un logotipo (ESPE), representando de tal manera un escenario en 3D el cual dentro de la ventana de jerarquías de Unity tiene el nombre de “Plane”, a ese *GameObject* se ha incorporado un script llamado “Rotar Escenario” permitiendo así manipular el componente *transform* y modificar el parámetro *Rotation* para hacer girar el escenario alrededor de la cámara que visualiza el menú (ver figura 22). El código que permite este evento se ejecuta dentro del método *Update ()* para que el valor siempre se actualice, la instrucción que se usa es la siguiente:

```
transform.Rotate (new Vector3(0,0.5f,0));
```



Figura 22 Fondo de escenario "menú principal"

Frente a la cámara y el controlador Leap Motion virtual se crea un *canvas* en el cual se dibujaran todos los menús que interactuaran con el usuario, para ver cómo preparar una escena en la que se interactúa con elementos UI mediante Leap Motion (revisar Anexo 2). Se incorporaran 4 interfaces GUI (1) Menú Principal con los botones correspondientes para conocer las opciones,

dar información de cada juego, ajustes generales y salida, (2) Información del juego Laberinto que contiene una descripción de las reglas y el objetivo del juego, (3) Información explorar museo que presenta al usuario la manera de usar los gestos de la mano para navegar por la escena museo y (4) pantalla con los justes de la resolución de la pantalla.

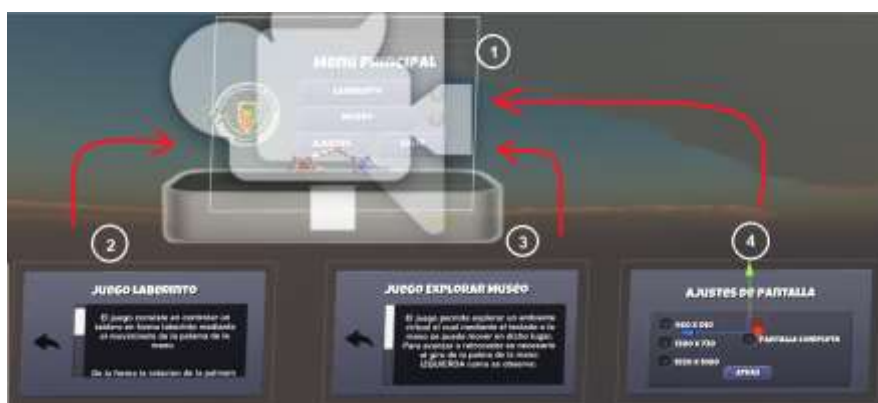


Figura 23 Interacción entre menús

Los GUI desarrollados cuentan con una animación para pasar de forma interactiva entre menús y finalmente se crea un *GameObject* el cual gestiona la escena, es decir en éste se añade dos scripts: el primero "Ajustes Control" para gestionar la resolución de la pantalla y el segundo "Cambio Escena" para controlar los cambios de escena al interactuar con los botones.

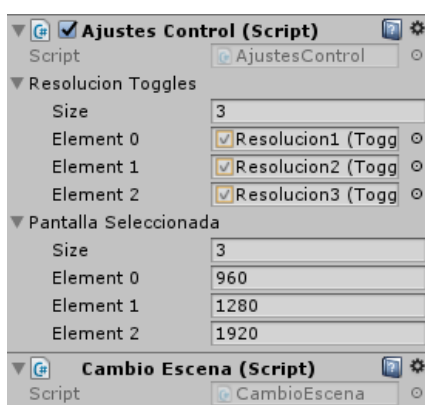


Figura 24 Componentes (script) en el objeto "GestorEscena"

3.3.2. Escena “Laberinto”

El desarrollo de esta escena fue pensada como un entorno de “habitación de juegos”, para esto se exporta a Unity una carpeta con distintos modelos de juguetes (ver figura 25) los cuales se ubican en un escenario. Estos objetos se colocan en la ventana de jerarquías del proyecto con sus respectivas coordenadas tridimensionales.

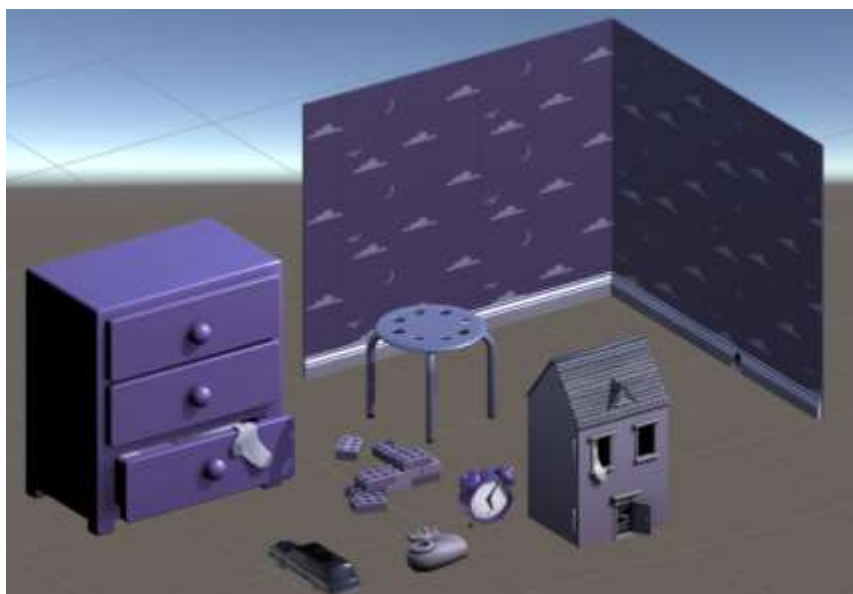


Figura 25 Modelos de juguetes para la escena

En el caso del tablero del laberinto se lo desarrolla en un software de modelado llamado Blender en el cual se puede crear del tipo y la forma que se desee (ver figura 26). Una que se haya escogido los modelos se los coloca en la escena, ubicados de forma que la cámara pueda captar todo el entorno.



Figura 26 Modelado del tablero "laberinto" en Blender

En la escena se incorpora nuevamente un *canvas* en el que se dibuja el menú del juego "Laberinto", el cual interactúa mediante las manos virtuales (revisar Anexo 2), dicho menú aparece en la escena cuando se haya presionado la tecla "Escape" para pausar el juego y se podrá acceder a las opciones que se observan en la figura 27. Para brindar un mejor detalle dentro del juego se incorpora elementos adicionales donde se observa el nivel del juego y la vida que le queda al jugador.



Figura 27 Menú del juego "Laberinto"

Después de ubicar todo en escena se procede a identificar el funcionamiento de la aplicación, se requiere que el tablero del juego se mueva conforme a la rotación de la mano (Anexo 1) para lo cual se elabora un *script* que cumpla con dicha función, a este código en C# se lo nombra como “**Engine_leap**” y como parámetro de entrada se ingresa un valor que permite la velocidad de giro del tablero, se considera un sistema de tres vidas el cual será proporcionado al inicio del juego, en este caso es la esfera que se mueve por el tablero, haya caído por cualquier agujero, el *script* que mantiene proporciona este control se lo ha designado “**Vida jugador**” donde se coloca el número de vidas con la que se empieza el juego y finalmente otro aspecto a considerar es cuando la esfera haya llegado a la meta, donde se ejecuta un código “**Ganar Nivel**” indicando la escena a la que el juego pasara una vez que se complete el escenario.

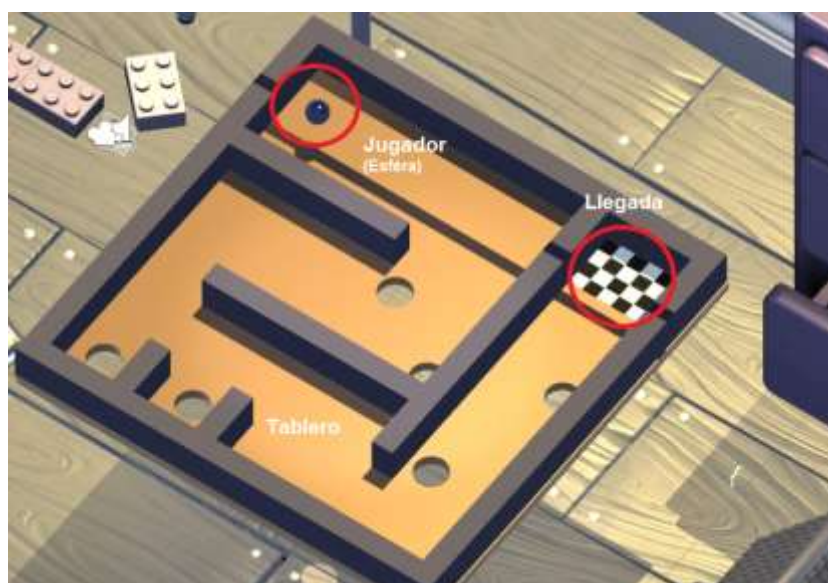


Figura 28 Elementos que componen el mecanismo del juego

El *script* “**Engine_leap**” es un código que hace uso de los ángulos de navegación *Pitch* y *Roll* según el movimiento de la mano del usuario para modificar los parámetros de rotación del componente *transform* del objeto “Laberinto” (el tablero), la validación para que se ejecute es mediante el uso

exclusivo de la mano derecha, es decir que el juego funcionara siempre y cuando el sensor Leap Motion identifique que se usa la mano derecha, caso contrario el tablero permanecerá en su posición original.

Luego de culminar con el nivel 1 el programa hace el cambio de escenario a un nivel 2 donde la dificultad aumenta ya que el tablero de laberinto tiene más huecos y un camino más largo (ver figura 29), la nueva escena contiene el mismo entorno de fondo prevalecen la mayoría de componentes, objetos, estructura y script.

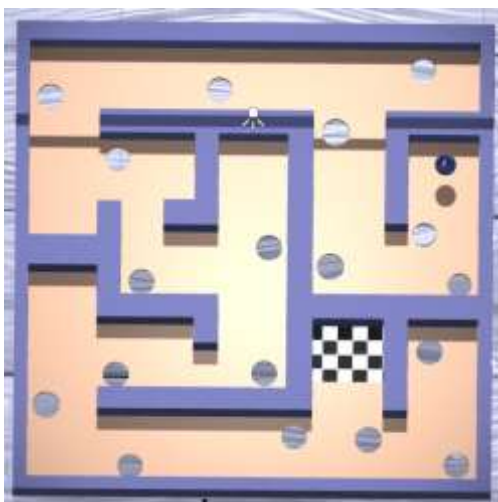


Figura 29 Tablero de laberinto nivel 2

3.3.3. Escena “Museo”

A esta escena se puede acceder a través del menú principal, presionando el botón Museo, el entorno desarrollado es una habitación la cual simula estar en un lugar para explorar la cultura de Quito, específicamente dos de las iglesias más importantes, en este escenario se encuentran diferentes objetos como son: cuadros, estante, maquetas de cada iglesia, botones en los que se puede acceder a material ilustrativo, para conocer más sobre la historia de la Catedral y la Basílica (ver figura 30).



Figura 30 Entorno del escenario "Museo"

La primera actividad que se puede realizar al explorar el “museo” es identificar la maqueta correspondiente a cada cuadro ubicado en las paredes: si por ejemplo nos encontramos en la iglesia de la Basílica, se debe dirigir al estante donde se encuentran las iglesias en pequeña escala e identificar qué modelo corresponde a la iglesia de la Basílica, mediante el uso de *Interaction Engine* (motor de interacción, revisar Anexo 2) se agarra la maqueta mediante un gesto de pinza con las manos y se dirige hacia el pequeño pilar que se encuentra cerca de la pintura para ubicarla en su respectivo lugar.



Figura 31 Ubicación de objetos para la actividad 1

La siguiente actividad que se puede realizar presionar el botón “Mirar” (ver figura 32) para proceder a reproducir un video informativo sobre la historia y datos importantes de la iglesia, en cualquier momento de la reproducción se

puede presionar en el botón atrás para terminar parar el archivo multimedia y seguir navegando por el entorno.



Figura 32 Menú de actividades (iglesia Catedral)

La escena de “Museo” está desarrollada como un juego de FPS por lo cual para navegar por el lugar existen dos opciones, utilizar el teclado (flechas o teclas A, W, S, D) para ir hacia adelante-atrás, izquierda-derecha, o emplear los movimientos exclusivamente con la mano izquierda (ver figura 33), mientras que con la mano derecha se realiza la acción de atrapar o coger objetos.

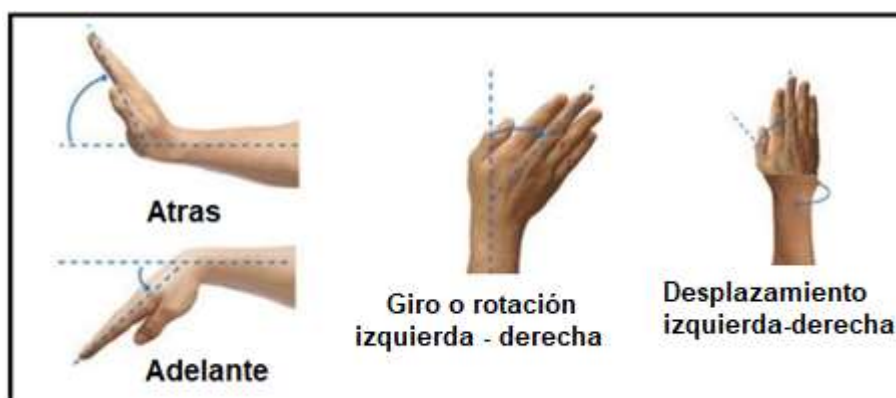


Figura 33 Movimientos necesarios para navegar la escena

La programación en cuanto a los menús (inicio, ver figura 34 e iglesias) tiene la misma ejecución como en las escenas descritas anteriormente, el

código más importante para la creación de la aplicación “Museo” son los *scripts* “**Control Cam Leap**”, “**Menú Inicio Manager**”, “**Controlador Trigger**” y “**Reproducir Video**”



Figura 34 Menú inicio de la escena "Museo"

Control Cam Leap: verifica si la mano captada por el sensor Leap Motion es la izquierda, una vez validado captura los ángulos de navegación (*Roll, Pitch, Yaw*) para mover al “personaje” por la habitación. **Menú Inicio Manager:** verifica si la tecla “escape” ha sido presionada para mostrar el menú (ver figura 34), se encarga pausar la escena para acceder a los botones donde se puede continuar en la aplicación, regresar al menú principal o salir del programa. **Controlador Trigger:** se encarga de conocer la ubicación de las maquetas, controla la animación si se encuentra en el estante principal, verifica una vez que haya caído al suelo ubicar los modelos en su posición inicial o la iglesia ha llegado al pilar correspondiente. **Reproducir Video:** es el encargado de ejecutar la reproducción del video informativo de las iglesias sobre un UI.

Finalizado el desarrollo de los escenarios se procede a seleccionar aquellos con las que se hace la construcción de la aplicación final (ver figura

35), es importante conocer el orden y la numeración en que se ubica cada aplicación, ya que dicho número se ingresa en los scripts que permiten gestionar los cambios de escenario, aquel que tenga índice cero será el primero en ejecutarse una vez que se construya y corra el programa.

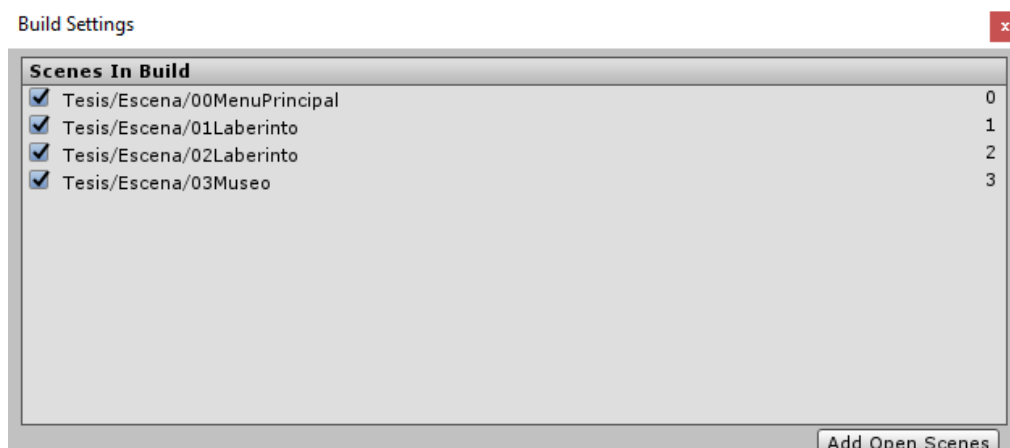


Figura 35 Escenas a ser construidas en la aplicación final

CAPITULO 4

4. PRUEBAS Y RESULTADOS

4.1. Pruebas del sistema

4.1.1. Evaluación del funcionamiento del sensor

Es necesario realizar pruebas respecto al funcionamiento del sensor Leap Motion, de manera que se evalué distintos aspectos que limiten su rendimiento, los apartados más importantes para considerar son la luz ambiente a la que el sensor puede funcionar y el campo de visualización para ejecutar los gestos que funcionan sobre la aplicación. Como se especifica en el capítulo 2, se puede verificar que la superficie de visión y de interacción donde la mano se puede movilizar es adecuada (ver figura 36) y no presenta algún retardo con la captura de cada fotograma.

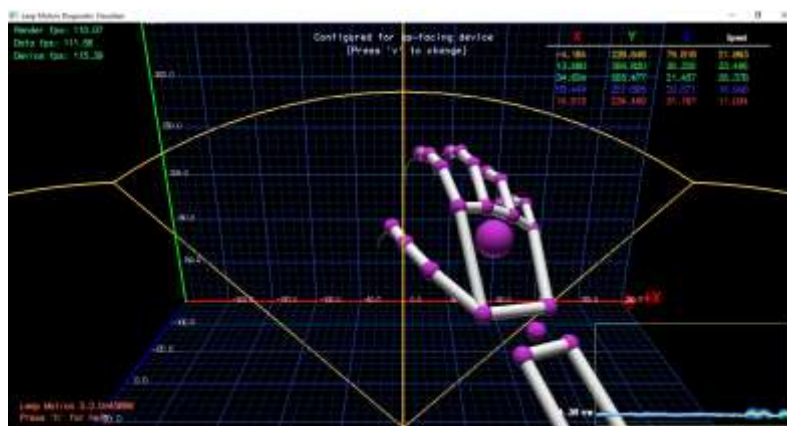


Figura 36 Funcionamiento en campo de visualización del sensor

La iluminación ambiente es un aspecto importante a la hora de probar el funcionamiento del sensor, para lo cual se ha considerado tres variaciones de

luz: clara natural, tenue artificial y nocturna sin iluminación. Los resultados muestran que Leap Motion reacciona adecuadamente a los distintos tipos de ambiente. Según la norma ISO 8995 la condiciones adecuadas para trabajar y desempeñar tareas deben contar con ambientes de luminosidad en rango de 900 a 1000 luxes, punto en el cual el sensor también funciona de forma adecuada.

4.1.2. Evaluación del funcionamiento de la aplicación

Las pruebas para evaluar la aplicación permiten resolver los posibles conflictos que se pueden presentar con el usuario, como primer punto es necesario la ubicación correcta del hardware (computador y sensor) para empezar a navegar por los distintos escenarios (ver figura 37).



Figura 37 Ubicación del *Hardware*

4.2. Pruebas de concepto

4.2.1. Pruebas de usabilidad

Para conocer los resultados sobre el impacto y usabilidad del sistema se realiza una ficha de evaluación que se aplica al usuario inmediatamente después de que este haya probado y usado la aplicación. Para la elaboración

de esta ficha se toma de referencia el cuestionario SUS, las métricas de prueba de usabilidad basadas en la norma ISO / IEC 9126-4, el manual de encuestas de satisfacción al consumidor y la heurística de NIELSEN (ver figura 38).

Los aspectos a considerar en el sistema de evaluación son:

- Lenguaje entendible y legible para el usuario
- Facilidad de usabilidad del usuario
- Visualización clara y legible de las interfaces
- Eficiencia de uso del sistema
- Métricas de eficiencia

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

Ficha de evaluación del sistema de entrenamiento físico mediante el sensor Leap Motion Controller

Evaluador

| Preguntas | Observaciones |
|---|---------------|
| ¿El usuario conoce el sensor previamente? | No |
| ¿Qué porcentaje de temas pudo completar el usuario? | 80% |
| ¿Qué tiempo empleó el usuario en completar todo el sistema? | 5 |

En esta encuesta se realiza con la finalidad de conocer su opinión sobre el uso del sistema de entrenamiento.
Marque con una X la respuesta que considere

Usuario

Sexo: M F Edad: 23

| Pregunta | Si | No |
|---|-------------------------------------|--------------------------|
| 1 ¿La interfaz muestra los resultados de sus pruebas y navegación? | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 2 ¿La interfaz ofrece ayuda o información sobre cómo usar el sistema? | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 3 ¿Se sintió cómodo usando el sensor y el sistema? | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 4 ¿El lenguaje de la aplicación "Entrenor" es claro y bien definido? | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 5 ¿El objetivo de la aplicación "Misión" es claro y bien definido? | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

| Preguntas | Muy bajo | Bajo | Medio | Alto | Muy alto |
|--|-------------------------------------|-------------------------------------|-------------------------------------|------|-------------------------------------|
| 1 Encuentro el sistema complejo y muy difícil de usar | | <input checked="" type="checkbox"/> | | | |
| 2 Encuentro un espacio visual muy sobrecargado (textos y elementos gráficos) | | | <input checked="" type="checkbox"/> | | |
| 3 Necesito leer el manual para usar correctamente el sistema | | <input checked="" type="checkbox"/> | | | |
| 4 Encuentro este sistema interesante y motivador | | | | | <input checked="" type="checkbox"/> |
| 5 Encuentro este sistema aburrido y poco útil para los deportes | <input checked="" type="checkbox"/> | | | | |
| 6 Este sistema sería como herramienta para entrenar y aprender | | | | | <input checked="" type="checkbox"/> |

Figura 38 Ficha de Evaluación del sistema

4.3. Análisis de resultados

- **Muestra**

La muestra utilizada para realizar las pruebas de usabilidad es de una población variada en edad y género, con estudios universitarios y

conocimientos sobre tecnología, ya que como se había indicado el sistema está enfocado para jóvenes-adultos de cualquier área de conocimiento, sin embargo las aplicaciones no están exentas de ser usado por niños quienes aprender mediante el juego y se relacionen de manera favorable con la tecnología. En la tabla 5 se puede observar los usuarios que utilizaron el sistema.

Tabla 5
Muestra utilizada para evaluar el sistema

| Edad | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 32 | 39 | Total general |
|---------------|----|----|----|----|----|----|----|----|----|---------------|
| Genero | | | | | | | | | | |
| Femenino | | 6 | 3 | 2 | | 1 | | | | 12 |
| Masculino | 1 | 3 | 13 | 11 | 2 | | 1 | 2 | 1 | 34 |
| Total general | 1 | 9 | 16 | 13 | 2 | 1 | 1 | 2 | 1 | 46 |

Al realizar la tabulación de los datos de la ficha desarrollada (Anexo 3) se puede analizar una de las métricas de usabilidad, la cual mostró qué porcentaje de personas conocen el sensor con el que van a trabajar, al ver la figura 39 se observa que aquellos que conocen al dispositivo “medio” (2 personas) solo saben de su funcionamiento pero nunca lo han utilizado y más del 50% (38 personas) no conocían ni la existencia de Leap Motion Controller lo que implica que el proyecto desarrollado es interesante para el usuario con nuevas aplicaciones que atraigan su interés.

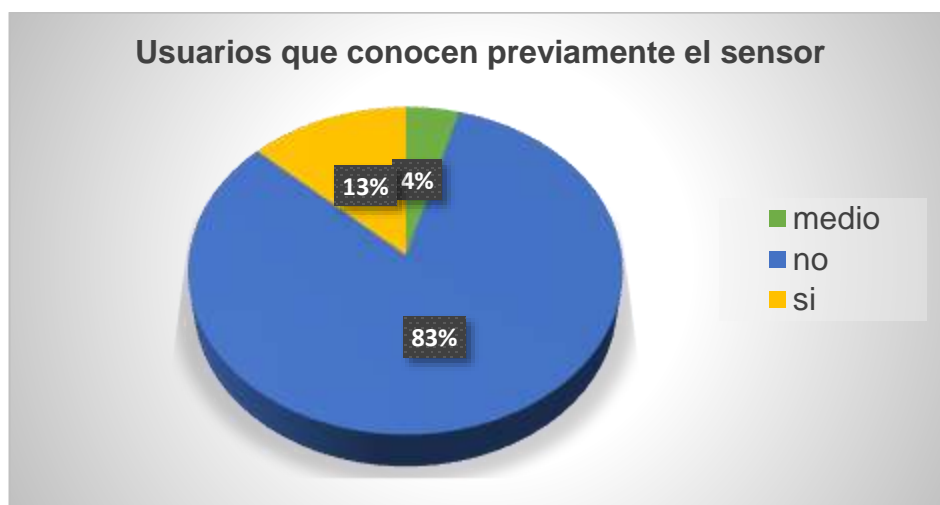


Figura 39 Porcentajes de personas que conocen el sensor

Se ha determinado un tiempo máximo de 10 minutos para que los usuarios manipulen el sistema de forma que se podrá evaluar dos métricas de usabilidad, es decir el tiempo de uso del sistema para conocer el porcentaje de tareas que se completó del sistema, en la figura 40 se observa una gráfica de la relación entre las dos métricas a evaluar, las medias de estos valores analizados son: 5 minutos y 10 segundos para cada usuario completando la navegación por todo el sistema de un 67%.

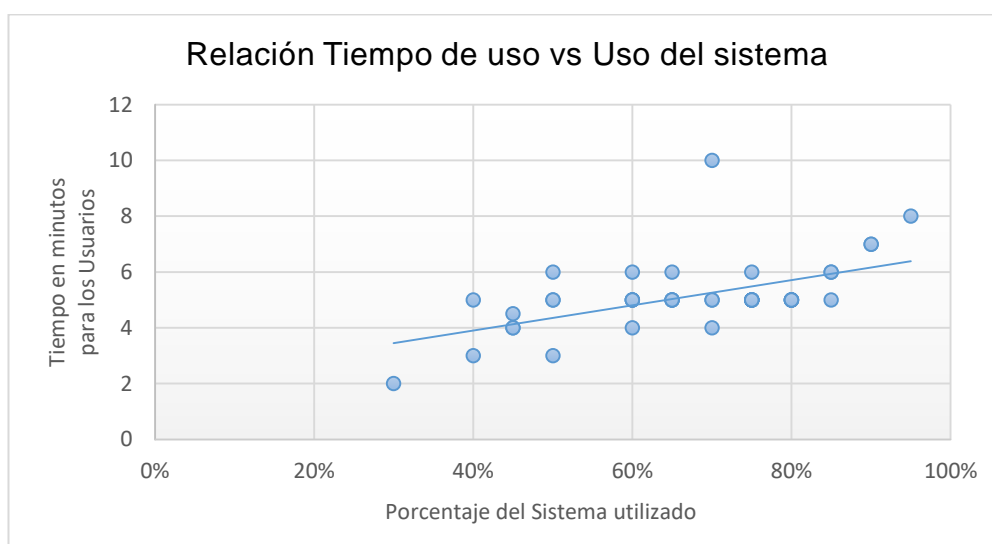


Figura 40 Porcentaje usado de sistema vs Tiempo empleado

La estimación del sistema se estructura de dos formas: conocer la aceptación de los evaluados hacia la interfaz de usuario ofrecida, y la calidad de todo el sistema para su usabilidad, lo cual se presenta dos tablas en la ficha de evaluación (Anexo 3). A continuación se muestra los resultados en preguntas cerradas de Si-No.

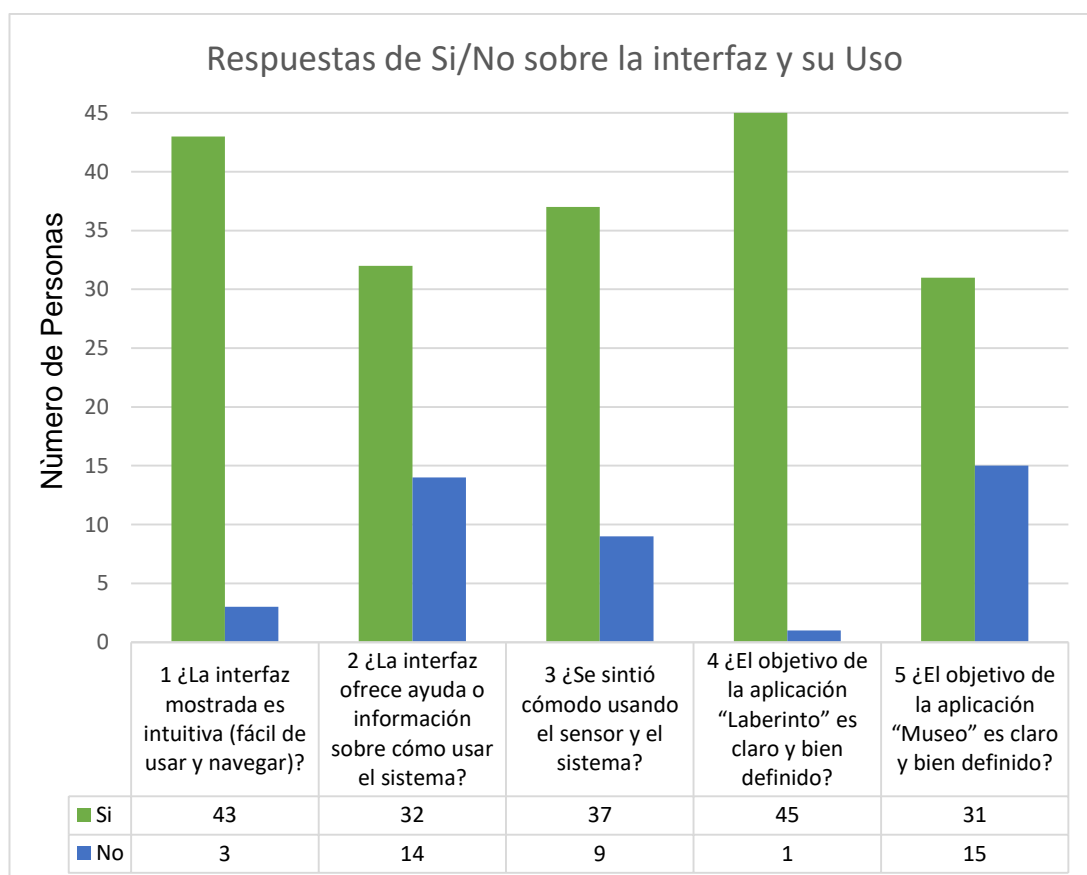


Figura 41 Resultados de los usuarios sobre la interfaz del sistema

Con los datos mostrados en la figura 41 se puede indicar que la interfaz en general del sistema es agradable al usuario ya que en su mayoría les pareció fácil de usar y navegar. Analizando la pregunta número 2 y la pregunta número 5 se observa que existe un gran porcentaje de personas que respondieron "No". Para lo cual fue necesario, dentro de los menús de ayuda, incorporar un apartado visual más interactivo del funcionamiento y objetivo de cada aplicación para realizar las actividades que corresponden a cada

escenario, es decir que en el botón de ayuda no solo hay texto que instruya al usuario a manejar las aplicaciones sino también imágenes animadas para indicar las acciones que pueden ejecutarse.

Al definir un límite de tiempo para que los usuarios utilicen el sistema, al primer grupo de evaluados no se dio indicaciones de cómo debían usarlo ya que es necesario evidenciar el desenvolvimiento de las personas para observar los errores más comunes que se presenten, posteriormente se corrigió algunos fallos para mejorar las acciones de los usuarios, al resto de evaluados se indicó brevemente las pautas necesarias para utilizar cada aplicación, mismos pasos que se encuentra en el manual de usuario (Anexo 1), comparando (ver figura 42) estos gráficos se puede indicar en la figura 42.b existe un mejor desempeño en el uso de la interfaz de usuario.

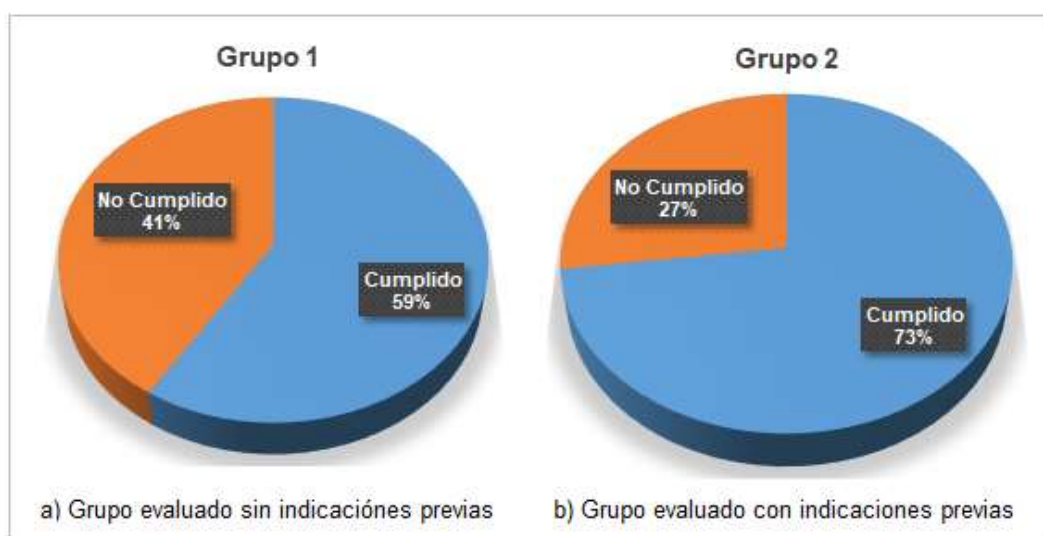


Figura 42 Porcentaje de desempeño en dos grupos de evaluación

En la segunda forma de la evaluación se realizó un análisis completo del sistema, se presentaron 5 niveles de satisfacción para que el usuario evalúe, de lo cual se obtuvo los siguientes resultados (ver tabla 6). Las preguntas 1, 2 y 5 muestran cualidades negativas por lo que una calificación máxima sobre el sistema se considera el nivel "Muy baja".

Tabla 6

Número de usuarios evaluando nivel de satisfacción, pregunta 1, 2 y 5

| Enunciado | Muy baja | Baja | Media | Alta | Muy alta |
|---|----------|------|-------|------|----------|
| 1 Encuentro al sistema complejo y muy difícil de usar | 6 | 9 | 28 | 3 | 0 |
| 2 Encuentro un apartado visual muy sobrecargado (interfaz y escenarios ilegibles) | 71,74% | | 6 | 6 | 1 |
| 5 Encuentro este sistema aburrido y poco útil para los demás | 95,65% | | 2 | 0 | 0 |

En la pregunta 1 de la segunda parte de la evaluación desarrollada se observa que en cuanto a la usabilidad del sistema, los usuarios la encuentran moderadamente difícil y complejo de usar completamente, es decir que para el uso tanto del sensor como de la interfaz gráfica de usuario, es necesario que 28 personas de 46 (ver figura 43) lean previamente un manual y experimenten más tiempo con el Leap Motion Controller.

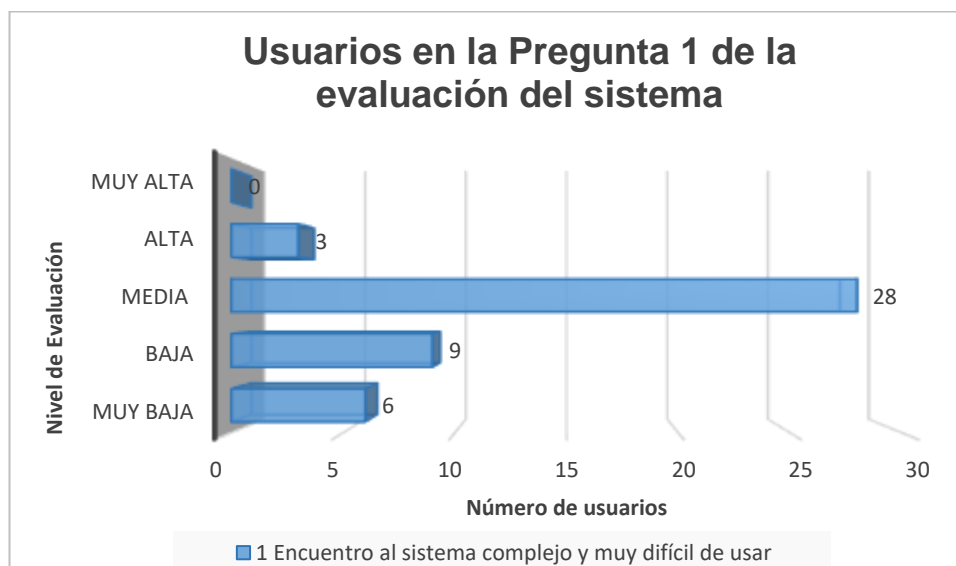


Figura 43 Tabulación sobre la dificultad del uso del sistema

Se puede ver que para la pregunta 2: ¿Encuentro un apartado visual muy sobrecargado (interfaz y escenarios ilegibles)? Los resultados entre “Muy Baja” y “Baja” superan el 70% indicando claramente que la interfaz del sistema

es adecuada y llamativa para el usuario. Mientras que en la pregunta 5: ¿Encuentra este sistema aburrido y poco útil para los demás? El 95.65% indica que el sistema no es aburrido (ver tabla 6).

Para el análisis de las preguntas 4 y 6 el nivel de satisfacción de usuario máxima con el que se puede evaluar el sistema es con “Muy Alta” ya que se resalta cualidades positivas del sistema sobre los usuarios, indicando si es una herramienta de entretenimiento y aprendizaje efectiva hacia los demás, y la pregunta número 3 es la percepción que cada usuario tiene para manipular el sistema sea o no con ayuda del manual de usuario, dependiendo únicamente de sus propias capacidades, por lo que los resultados obtenidos se pueden observar a continuación (ver figura 44).

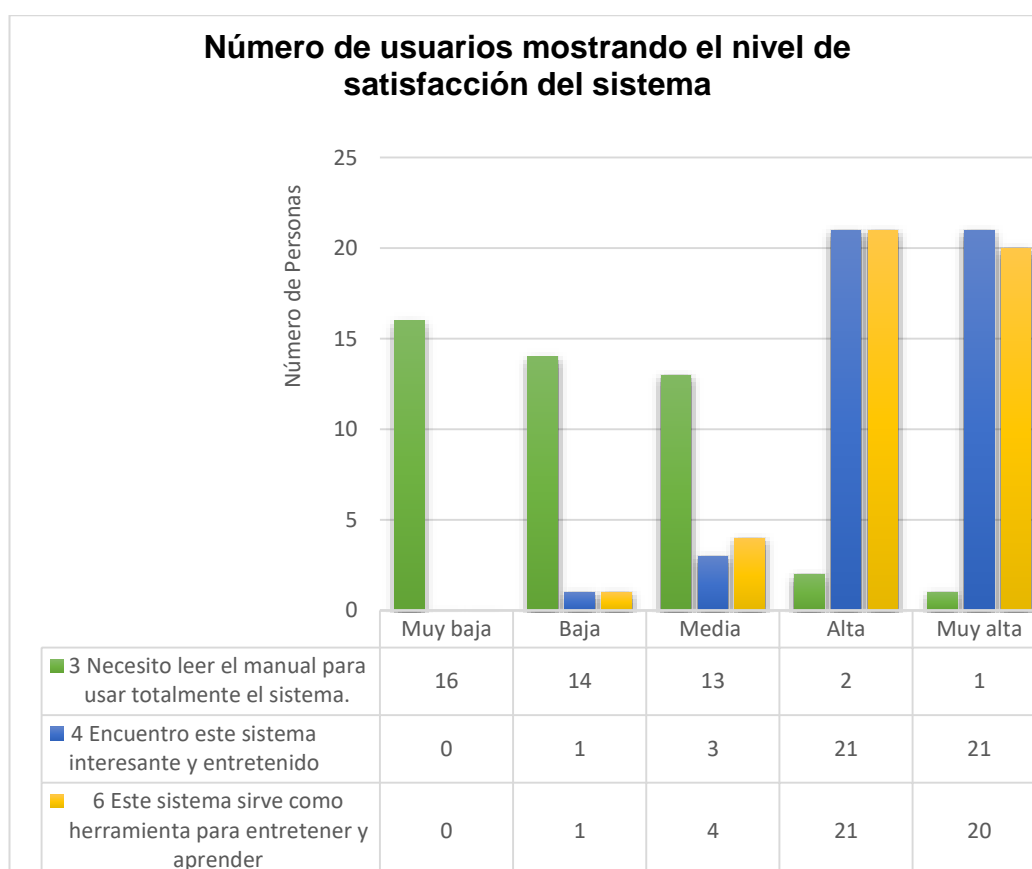


Figura 44 Usuarios evaluando el nivel de satisfacción pregunta 3, 4 y 6

El resultado obtenido de la pregunta número 3, se puede observar que un porcentaje de 34.78% (16 personas, “Muy Baja”) no necesitan usar el manual de usuario para manipular el sistema, el 30.43% requieren de una revisada mínima para solventar alguna duda y el 28,26% condicionan a un mejor uso del sistema una leída de la documentación, como se puede analizar el mayor porcentaje de encuestados se encuentran en los niveles bajos de satisfacción lo que indica que el sistema es intuitivo y fácil de usar, permitiendo que el manual de usuario sea un complemento del sistema.

Las preguntas 4 y 6 se basan en analizar si el usuario identifica al sistema como una herramienta tanto de entretenimiento como de aprendizaje por lo que mediante los datos obtenidos en las encuestas se realiza una gráfica de tendencia para conocer si el nivel de satisfacción al usuario es el más alto (ver figura 45).

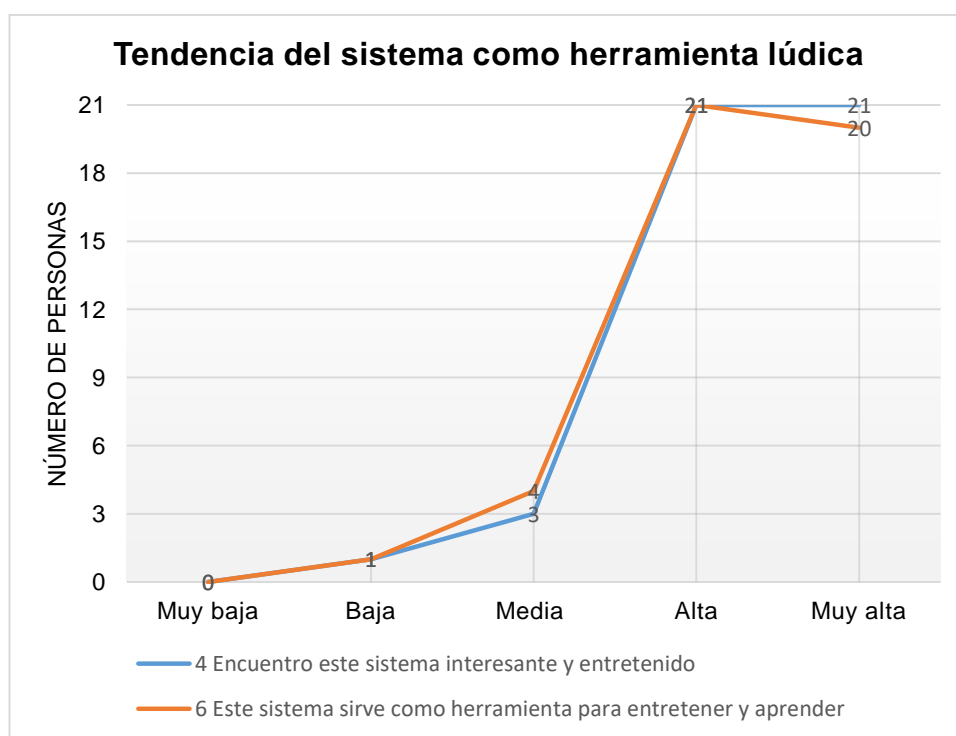


Figura 45 Tendencia del sistema como herramienta de entretenimiento y aprendizaje

La pregunta 4 describe si el usuario encuentra interesante y entretenida a las aplicaciones por lo que se puede observar que el 91.30% (42 usuarios) indican que si valorando el nivel entre “Alta” y “Muy Alta”. Mientras que en el enunciado 6 se indica al usuario si el sistema es una útil herramienta para el aprendizaje indicando de igual manera los niveles más alto con un 89.13% (41 encuestados) de aceptación.

Después de la observación al usuario usando la aplicación se logró corregir los siguientes aspectos relevantes en la interfaz de usuario para el mejor desempeño del sistema.

- Errores semánticos, información incompleta.
- Errores ortográficos y tipográficos.
- Errores de validación, aspectos donde el juego no realizaba alguna acción o validación.
- Facilidades de uso y entendimiento de la aplicación.

CAPITULO 5

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

El incorporar dispositivos HCI en herramientas didácticas y de aprendizaje permite mejorar ampliamente la experiencia de usuario permitiéndoles explorar distintas maneras en las que pueden manipular dispositivos que previamente no han utilizado, por consiguiente el sistema tiene una mayor aceptación como producto final.

Como el sensor Leap Motion Controller provee un flujo de datos adecuado, se puede visualizar en tiempo real el movimiento de las manos dentro del entorno virtual, permitiendo al usuario acercarse a distintos campos de aplicación como la realidad virtual y medicina en rehabilitación de la motricidad fina.

La configuración y ubicación de las cámaras infrarrojas que detectan las manos permiten crear una amplia zona de trabajo en donde el usuario puede interactuar, logrando cubrir una superficie tridimensional semiesférica de 61 cm de radio y una apertura de ángulo un poco menor a los 150° que varía según la intensidad de luz que afecte al dispositivo.

Cuando las manos virtuales que se observan en la interfaz gráfica experimentan un LAG¹ al momento de usar las aplicaciones las causas se deben a la exposición prolongada del uso del sensor ya que este tiende a

¹ Retraso producido en una telecomunicación que dificulta el desarrollo normal de la misma.

recalentarse, o que el dispositivo pierde lectura de las manos del usuario cuando no se encuentra en el campo visible del Leap Motion.

En el desarrollo de las aplicaciones, se debe considerar todos los tipos de movimientos que pueden realizar las manos sobre el sensor ya que cuando la palma de la mano se encuentra perpendicular al dispositivo existen conflictos en la lectura de datos y estos se los debe manejar adecuadamente para que el sistema funcione de manera correcta.

Los botones de ayuda son una fuerte herramienta para entender el sistema previo a su uso, sin embargo la implementación de un asistente de instrucciones (como los que existen en los videojuegos), en el inicio de cada juego es importante ya que estos permiten ir aprendiendo las acciones accesibles al usuario para realizar mientras se avanza paso a paso.

La arquitectura de software con la que se maneja Unity permite desarrollar aplicaciones sencillas con alto impacto visual ya que al trabajar el código en C# como componentes de un objeto, éste permite caracterizarlo de forma única dentro de los escenarios, así es posible crear *scripts* menos complejos y con alto grado de abstracción.

Las herramientas lúdicas y de entretenimiento tienen mayor aceptación de los usuarios cuando brindan mejor experiencia de usabilidad, por lo que mediante el sensor Leap Motion Controller es posible manipular objetos dentro de la aplicación gracias a la física de objetos que provee Unity y el motor de interacción que brinda las librerías del sensor.

Los resultados que muestran la pregunta sobre si el sistema es complejo y difícil de usar, el 60% de los usuarios indica un valor medio en la complejidad para manipular el sistema, esto se debe principalmente a que el

83% de los evaluados nunca habían usado el sensor Leap Motion Controller lo que implica que inicialmente va a ser difícil manejar del sistema.

5.2. Recomendaciones

Es recomendable verificar las versiones de cada software, tanto del programa de Unity como los paquetes de instalación del sensor Leap Motion Controller ya que pueden causar problemas de funcionamiento al momento de conectarlo a la computadora e intentar utilizar el sistema de entretenimiento.

Revisar constantemente la página oficial de Leap Motion ya que en ella los desarrolladores y comunidad actualizan los paquetes que permiten usar el sensor dentro de la plataforma de Unity, seguidamente se cambian o modifican los módulos y al momento de cargarlos dentro del proyecto pueden existir conflictos de versiones o códigos obsoletos.

Es importante revisar las características técnicas del computador que se use, ya que para el correcto funcionamiento del sensor es necesario contar mínimo con puertos USB 2.0 en adelante y adecuar correctamente la intensidad de luz que incida sobre el sensor en el lugar que se vaya a utilizar el sistema .

Otro aspecto importante en la característica de la computadora es conocer la potencia de la tarjeta gráfica ya que para tener una mejor experiencia en la calidad de los gráficos es recomendable tener un procesador i3 en adelante que permita renderizar las escenas de la aplicación, caso contrario se observara imágenes pixeladas.

Es recomendable consultar la página y documentación oficial de Unity ya que en esta se indican los cambios de la plataforma en cuanto a sintaxis

del lenguaje de programación al igual que los formatos y extensiones de archivos permitidos para crear cualquier tipo de aplicación, por ejemplo es necesario importa videos vía Apple QuickTime² para lo cual se requiere tenerlo instalado en el computador.

Al momento de usar la aplicación es recomendable revisar los gestos de las manos que el sensor Leap Motion capta, ya que el movimiento adecuado para generar la interacción con los menús dentro de los entornos del sistema es el gesto de *Screen Tap Gesture*, que permite simular una pantalla frente a los usuarios.

5.3. Trabajos Futuros

Los análisis de resultados muestran al sistema implementado como una gran herramienta lúdica por lo que es posible dar continuidad a las aplicaciones desarrolladas, de manera que se pueda incorporar escenarios mucho más amplios, es decir que se puede crear, mediante Unity, ambientes virtuales más elaborados y extensos donde los usuarios puedan navegar extensamente como en los videojuegos, al igual que añadir niveles en las aplicaciones propuestas.

Otra de las líneas de investigación para trabajos futuros, es incorporar en las aplicaciones desarrolladas las gafas VR³ como son las de Oculus Rift o HTC Vive⁴, ya que al ser Unity multiplataforma existe mayor facilidad de migrar el sistema a entornos de Realidad Virtual (VR) de manera que se consiga crear sistemas más inmersivos para los usuarios.

² Framework (entorno de trabajo) multimedia estándar compatible con videos MPEG-4

³ Entorno de escenas u objetos de apariencia real

⁴ Cascos de realidad virtual

Se puede ampliar la investigación del trabajo presentado para corregir errores de lectura del Leap Motion, es decir implementar un sistema de dos sensores (A y B) los cuales se encuentren perpendiculares entre sí, el sensor "A" capturará los datos de la mano al encuentre paralelo a este, mientras que el sensor "B" capturará los datos cuando la mano se encuentre perpendicular al sensor "A" de forma que no se pierdan la lectura de datos sin importar la rotación realizada por la mano del usuario.

REFERENCIAS BIBLIOGRAFICAS

Alemañ Baeza, T. (2015). Desarrollo de un videojuego para móviles con Unity. Universidad de Alicante .

Alfaro Vives, D. A., & Velarde Robles, D. M. (2015). *Desarrollo de aplicaciones con leap motion*. Lima: Universidad Peruana de Ciencias Aplicadas (UPC).

Assis de Castro, R. H., Soares Medeiros, A. C., & Aires Tavares, T. (2012). Natural Interaction for 3D Manipulation in Telemedicine: A Study Case developed for Arthron Video Collaboration Tool. 6.

Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. USA.

Belda, J. (20 de 04 de 2015). *Blog de ShowLeap*. Obtenido de LEAP MOTION (I): CARACTERÍSTICAS TÉCNICAS: <http://blog.showleap.com/2015/04/leap-motion-caracteristicas-tecnicas/>

Belda, J., & Montalvo Martínez, M. (04 de 05 de 2015). *Blog de ShowLeap*. Obtenido de LEAP MOTION (II): PRINCIPIO DE FUNCIONAMIENTO: <http://blog.showleap.com/2015/05/leap-motion-ii-principio-de-funcionamiento/>

Beltramone, D. A., Rivarola, M. F., & Quinteros Quintana, M. L. (2015). Interfaces táctiles aplicadas en el proceso de aprendizaje en la educación especial. *Virtualidad, Educación y Ciencia*, 13.

Beltramone, D. A., Tula, S. M., Rivarola, M. F., Hidalgo, M. B., Tancredi, P. D., & Quinteros Quintana, M. L. (2014). En búsqueda de Interfaces Naturales para Personas con Discapacidad. *IEEE Biennial Congress of Argentina (ARGENCON)*, 6.

C2Estudio. (s.f.). *Tutorial de introducción a Unity Integración de un proyecto de videojuego*. Obtenido de <http://trinit.es/unity/documentacion%20proyectos/lecturas/Tutorial%20de%20Introduccion%20a%20Unity3D.pdf>

Cabello Espinosa, M. E., & Ramos Salavert, I. (2016). *Ingeniería de aplicaciones dirigida por modelos y basada en técnicas de líneas de productos software*. Colima: Universidad de Colima.

Cepeda Díez, J., Meijome Sánchez, X. M., & Santillán García, A. (2012). Innovaciones en salud y tecnologías: las cosas claras. *Revista Enfermería CYL*, 5.

Colgan, A. (2 de Agosto de 2016). *Leap Motion Blog* . Obtenido de Unity Core Assets 101: How to Start Building Your VR Project: <http://blog.leapmotion.com/unity-core-assets-101-start-building-vr-project/>

Davis, A. (16 de 08 de 2014). *Leap Motion Blog*. Obtenido de Introducción al SDK de Leap Motion: <http://blog.leapmotion.com/getting-started-leap-motion-sdk/>

Developer, L. M. (s.f.). *Leap Motion Developer* . Obtenido de Setting up an Interaction Engine Scene: https://developer.leapmotion.com/documentation/v2/unity/unity/Unity_IE_Setup.html

Developers, L. M. (11 de Junio de 2016). *Leap Motion Blog*. Obtenido de New Unity Module for User Interface Input: <http://blog.leapmotion.com/ui-input-module/>

Disqus. (s.f.). *Leap Motion Developers*. Obtenido de Coordinate Systems: https://developer.leapmotion.com/documentation/v2/csharp/devguide/Leap_Coordinate_Mapping.html

Durán Muñoz, F., Troya Castilla, J., & Vallecillo Moreno, A. (s.f.). *Desarrollo de software dirigido por modelos*. Catalunya: Openlibra.

Escobar, G., Brey, G. A., & Martins, U. (2017). *Arquitecturas de Software*. Buenos Aires: Universidad Tecnológica Nacional Facultad Regional Buenos Aires.

Fernández Arroyo, A. (Septiembre de 2013). Obtenido de Simulador De Accidentes De Tráfico Mediante Motor De Videojuegos Unity: <http://repositorio.upct.es/bitstream/handle/10317/3474/pfc5512.pdf;jsessionid=52C642BEDAEF086F3A77BCB0CBBE6B07?sequence=1>

Fernández, P., Navarro, F., & Muñoz, Á. (2013). *LEAP MOTION*.

Freire Conrado, A. B., Pazmiño Moya, V. D., De La Cruz, D., & Tobar, J. (2015). Gesture Recognition Sensor: Development of a tool with playful applications to evaluate the physical and cognitive skills of children through the use of bootstrap algoritm. En A. Sio-long, K. Haeng , & A. Mahyar A., *Transactions on Engineering Technologies* (pág. 585). San Francisco, CA: Springer.

García Garcés, H., Navarro Aguirre, L., López Pérez, M., & Rodríguez Orizondo, M. d. (2014). Tecnologías de la Información y la Comunicación en Salud y Educación Médica. *SciELO Cuba*, 9.

González Huerta, J. (2014). *Derivación, Evaluación y Mejora de la Calidad de Arquitecturas Software en el Desarrollo de Líneas de Producto Software Dirigido por Modelos*. Valencia.

González Sotomayor, I. S. (2014). *Diseño y desarrollo de un videojuego educativo con técnicas de inteligencia artificial para la plataforma android aplicando la metodología OOHDM*. Sangolquí: Espe.

Guna, J., Jakus, G., Pogačnik, M., Tomažič, S., & Sodnik, J. (2014). An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. *sensor*, 19.

Karashanov, A., Manolova, A., & Neshov, N. (2016). APPLICATION FOR HAND REHABILITATION USING LEAP MOTION SENSOR BASED ON A GAMIFICATION APPROACH. *International Journal of Advance Research in Science and Engineering*, 9.

Kin , F. L., Sevcenco, A. M., & Yan, E. (2013). Telerehabilitation Using Low-Cost Video Game Controllers. *Seventh International Conference on Complex, Intelligent, and Software Intensive Systems*, 8.

Laazizi Ruiz, K. (2014). *Desarrollo de un prototipo usando como dispositivo de interacción Leap Motion*. Madrid.

Leap Motion Developer. (s.f.). *Leap Motion Developer*. Obtenido de API Overview:

https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html#motion-tracking-data

Losa, M., Morone, G., Fusco, A., Castagnoli, M., Fusco, F. R., Pratesi, L., & Paolucci, S. (2015). Leap motion controlled videogame-based therapy for rehabilitation of elderly patients with subacute stroke: a feasibility pilot study. *Topics in Stroke Rehabilitation*, 11.

Mendoza Ayo, D. A. (2016). *Diseño de la interfaz de control para un cuadricóptero con el dispositivo Kinect*. Sangolquí.

Ministerio de Educación. (2014). *Curriculum Educación Inicial 2014*. Quito.

Morales Cruz, V. A., & Tamayo Sánchez, W. A. (2017). *Diseño e implementación de un sistema con interacción gestual para enseñanza de lenguaje de señas básico ecuatoriano a niños con discapacidad auditiva mediante sensores gestuales*. Latacunga: Espe.

Nandy, A. (2016). *Leap Motion for Developers*. Kolkata, West Bengal : Apress.

Polo Usaola, M. (2013). *Desarrollo de software basado en reutilización*. Catalunya: Openlibra.

- Pons, C., Giandini, R., & Pérez, G. (2010). *Desarrollo de Software Dirigido por Modelos*. Buenos Aires: Edulp.
- Potter, L. E., Araullo, J., & Carter, L. (2013). The Leap Motion controller: a view on sign language. *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, (pág. 4).
- Quevedo Pérez, W. X. (2016). *Implementación de un esquema de teleoperación para un robot manipulador móvil maniobrado a través de dispositivos hápticos, para incrementar la transparencia del sitio remoto, a través de entornos de realidad virtual y realidad aumentada*. Latacunga.
- Secretaría Nacional de Planificación y Desarrollo. (2013). *Buen Vivir Plan Nacional*. Quito.
- Spiegelmock, M. (2013). *Leap Motion Development Essentials*. Birmingham - Mumbai: Packt Publishing Ltd.
- Texier, J., De Giusti, M. R., & Gordillo, S. (2014). EL DESARROLLO DE SOFTWARE DIRIGIDO POR MODELOS EN LOS REPOSITORIOS INSTITUCIONALES. *Servicio de Difusión de la Creación Intelectual (SEDICI)*, 10.
- Unity Technologies. (2016). *Unity Documentation*. Obtenido de Manual de Unity: <https://docs.unity3d.com/es/current/Manual/UnityManual.html>
- Vega Sosapanta , P. E. (2017). *Desarrollo e implementación de un sistema de apoyo académico usando instrucciones gestuales para niños mediante el uso del dispositivo Leap Motion Controller*. Sangolquí: Espe.
- Villacís, C., Fuertes, W., & Santillán , M. (2016). On the Development of Strategic Games based on a Semiotic Analysis: A case study of an optimized Tic-Tac-Toe. *SCITEPRESS - Science and Technology Publications, Lda* , 8.
- virtualwaregroup. (2013). *Virtual Rehab*. Virtual Ware.
- Yépez Montenegro, D. R. (2016). *Diseño e implementación del sistema de control de iluminación de una vivienda, controlado por el dispositivo leap motion, basado en realidad virtual*. Quito.