



Universidad de las Ciencias
Informáticas

Facultad 4

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Título: Componente para aplicar tecnologías de
realidad virtual a software con entornos 3D en
Unity

Autor: Lester Atreyu Reyes Capote

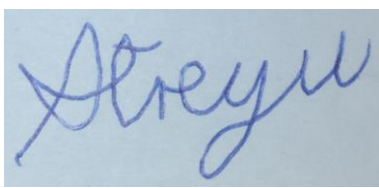
Tutor: Dr. C. Omar Correa Madrigal

La Habana, diciembre de 2022

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los 8 días del mes de diciembre del año 2022.



Lester Atreyu Reyes Capote

Autor



Dr. C. Omar Correa Madrigal

Tutor

Resumen

En el mundo se han desarrollado un gran número de aplicaciones y plataformas que usan la realidad virtual, desde sistemas para la educación y entrenamiento de habilidades, hasta juegos y plataformas para darle un enfoque novedoso a sus productos. En Cuba también se han desarrollado un gran número aplicaciones que usan realidad virtual, muchas de ellas realizadas por la Universidad de Ciencias Informáticas (UCI), de manera independiente y en colaboración con otras entidades como el Ministerio de Salud Pública (MINSAP).

En la actualidad, muchos sistemas de realidad virtual, pero en la Universidad de Ciencias Informáticas no es muy común su desarrollo. Sin embargo hay programas que permiten que aplicaciones con entornos virtuales sean consumidas como software de realidad virtual.

Por tanto, en la siguiente tesis se plantea como principal objetivo el desarrollo de un componente para aplicar tecnologías de realidad virtual a software con entornos en 3D en Unity.

Palabras clave:

Cardboard, Giroscopio, Interoperabilidad, Realidad Virtual, WIFI

Abstract

In the world, a large number of applications and platforms that use virtual reality have been developed, from systems for education and skills training, to games and platforms to give a new approach to their products. In Cuba, a large number of applications that use virtual reality have also been developed, many of them carried out by the University of Informatics Sciences (UCI), independently and in collaboration with other entities such as the Ministry of Public Health (MINSAP).

At present, there are many virtual reality systems, but at the University of Informatics Sciences their development is not very common. However, there are programs that allow applications with virtual environments to be consumed as virtual reality software.

Therefore, in the following thesis the main objective is the development of a component to apply virtual reality technologies to software with 3D environments in Unity.

Keywords:

Cardboard, Gyroscope, Interoperability, Virtual Reality, WIFI,

Índice

Introducción.....	7
Capítulo 1 Fundamentación Teórica	12
1.1 Introducción al Capítulo	12
1.2 Fundamentos de la Realidad Virtual	12
1.2.1 Realidad Virtual	12
1.2.2 Inmersión.....	13
1.2.3 Visión Estereoscópica	14
1.3 Análisis de aplicaciones homólogas	15
Trinus VR.....	15
VRidge.....	16
Firefox Reality.....	17
Intugame.....	18
Virtual Desktop	18
OpenVR Cardboard.....	18
1.4 Análisis comparativo de las aplicaciones homólogas	19
1.5 Método de ingeniería de Software	21
1.6 Herramientas y Tecnologías	21
1.6.1 Lenguaje del Modelado	21
1.6.2 Herramienta Case	22
1.6.3 Unity 3D.....	22
1.6.4 Visual Studio Code	23
1.6.5 C#.....	24
1.7 Conclusiones del capítulo	24
Capítulo 2 Análisis y Diseño.....	25
2.1 Introducción al capítulo	25
2.2 Propuesta de Solución.....	25

2.2.1	Diagrama de actividades	25
2.2.2	Requisitos Funcionales.....	26
2.2.3	Requisitos no Funcionales.....	27
2.2.4	Integración de ambiente RV en Unity	28
	XR Plug-in Magnement	28
	Mock HMD XR Plugin.....	29
	Multiplayer HLAPI.....	30
	Calibración del Giroscopio.....	31
2.3	Historias de Usuario.....	32
2.4	Plan de Iteraciones	35
2.5	Plan de Entrega	35
2.6	Tarjetas Clase-Responsabilidad-Colaboración	36
2.8	Estilo Arquitectónico	39
2.10	Conclusiones del capítulo	40
Capítulo 3 Implementación y Pruebas		41
3.1	Introducción al capítulo	41
3.2	Tareas de Ingeniería.....	41
	3.2.1 Iteración 1	41
	3.2.2 Iteración 2.....	42
	3.2.3 Iteración 3.....	43
3.3	Descripción de la Solución.....	44
3.4	Pruebas de Aceptación.....	44
	3.4.1 Iteración 1	44
	3.4.2 Iteración 2.....	45
	3.4.3 Iteración 3.....	46
	3.3.4 Resultados de las pruebas	47
3.4	Conclusiones del capítulo	48

Conclusiones generales	49
Recomendaciones	50
Referencias	51

Introducción

En el mundo hay una gran cantidad de avances tecnológicos orientados a mejorar la interacción hombre computadora y de brindar productos a los usuarios de maneras diversas y novedosas. En este contexto ha estado el tema de la realidad virtual, el cual ha ido evolucionando desde el siglo pasado. Hoy en día se puede viajar virtualmente a distintos lugares y sumergirse en estos entornos comiendo comida oriunda de estos lugares [1], aunque esto no se limita a espacios abiertos, también se han usado para visitar museos y otros lugares de ocio y aprendizaje [2]. Otros usos que se pueden ver es en el tema de la medicina, en la cual el Consejo Superior de Investigaciones Científicas en España ha logrado reducir los efectos del Parkinson en pacientes usando tratamientos que usan realidad virtual [3]. Vital importancia tiene el uso de estas tecnologías en la enseñanza, para que los alumnos puedan retener el conocimiento de una manera más visual y ayudar a los estudiantes con dificultad de aprendizaje, la cual es hasta usada en la rama militar para el entrenamiento de los soldados [4] y deportistas [5].

En Cuba también se hacen un gran número de trabajos referentes a la realidad virtual. Entre estos se encuentran:

- Experiencias en la creación de ambientes virtuales de aprendizaje desde un proyecto productivo del Polo de Realidad Virtual en la Universidad de las Ciencias Informáticas[6]
- Aprendizaje adaptativo basado en Simuladores de Realidad Virtual[7]
- Procedimiento de Arquitectura de Información para los entornos de Realidad Virtual con fines educativos[8]

Estas tecnologías sin embargo requieren de dispositivos de altas prestaciones para su uso por lo que este mercado está orientado para personas con un alto poder adquisitivo, ya que es necesario la adquisición de un oculus, google cardboard u otra tecnología parecida [6]. En la Ilustración 1 se pueden ver ejemplos de algunos dispositivos. Con el paso del tiempo el mercado se ha ido expandiendo y las prestaciones para estos dispositivos se han ido reduciendo



Ilustración 1

ligeramente, pero todavía no están al alcance de todo el público. Esta barrera se puede superar si se usan alternativas o buscando una oferta aceptable [9], [10].

Esto trae como consecuencia que en Cuba no se produzcan un gran número de softwares que utilicen la realidad virtual. Aun así, hay un gran número de juegos

que podrían adaptarse a esta tecnología o al menos incorporarle herramientas de realidad virtual. La mayoría de los videojuegos desarrollados por la Universidad de Ciencias Informáticas son realizados en la herramienta de desarrollo Unity, esto debe ser tomado en cuenta en su desarrollo.

A pesar de lo anterior planteado, hoy en día hay herramientas que permiten migrar entornos que se puedan generar en la pantalla de un monitor a uno de realidad virtual. Una de estas herramientas puede ser el Trinus VR [11], un software capaz de capturar la pantalla del monitor y dar la sensación de que cualquier escenario que se esté reproduciendo en esta sea el de un entorno de realidad virtual, bastante usada en estos casos [12], pero esta tecnología requiere de licencias privadas y su versión gratuita solo permite al usuario disfrutar de este entorno durante 10 minutos, lo que en determinados entornos sería muy restrictivo.

Teniendo en cuenta la problemática formulada anteriormente se plantea como problema de investigación, ¿Cómo proveerle a aplicaciones con entornos 3D en Unity, la capacidad para que puedan usar la Realidad Virtual?

Analizando lo anterior se puede obtener que el objeto de estudio: las aplicaciones que permitan la adaptación de entornos 3D a entornos de realidad virtual.

Identificando que el campo de acción se encuentra en los métodos, técnicas y herramientas que permitan a entornos 3D usar tecnologías de realidad virtual.

Teniendo como objetivo general desarrollar un componente para aplicar tecnologías de realidad virtual a software con entornos 3D en Unity.

A partir de lo anterior se proponen una serie de tareas de investigación, que son:

- Elaborar el marco teórico de la investigación a través del estudio del estado del arte para una mejor comprensión de la investigación.
- Identificar las técnicas a utilizar para desarrollar el componente para la adaptación de entornos 3D.
- Caracterizar las herramientas homólogas identificando características comunes y principales funcionalidades.
- Caracterizar las herramientas a utilizar en la solución propuesta y motivo de su uso.
- Generar artefactos ingenieriles en dependencia de la metodología utilizada.
- Programar las funcionalidades e interfaz de la herramienta según los artefactos utilizados.
- Validar la solución mediante el método científico determinado en la investigación.

Para la realizar las tareas pertinentes se tomaron una serie de métodos científicos:

- Revisión documental: para fundamentar el propósito de la investigación y permitir el desarrollo del marco teórico y conceptual de la misma.
- Analítico sintético: empleado a la hora de realizar el estudio de las teorías y documentos más relevantes sobre el uso e importancia de los sistemas capaces de a partir de un entorno 3D generar un sistema inmersivo de realidad virtual, permitiendo así, extraer los elementos más importantes sobre los mismos

- Modelación: usado para confeccionar el componente que pueda integrarse a diversos tipos de software para incorporar técnicas de VR de manera fácil y sencilla.

Además de estos se utilizarán también los métodos empíricos, destacando las entrevistas y encuestas. Las entrevistas tienen un papel fundamental al inicio del proyecto de investigación por permitir la interacción directa con los involucrados en la problemática permitiendo al investigador enfocar sus acciones de una forma más coherente en base a las soluciones.

Se desarrollaron preguntas fundamentales como:

- ¿Qué otros proyectos se han realizado en la universidad relacionados con la realidad virtual?
- ¿Qué herramientas se utilizan en la mayoría de las aplicaciones que usan técnicas 3D en la universidad?
- ¿Cuáles son las principales dificultades que se pueden encontrar a la hora de desarrollar un sistema relacionado con la realidad virtual?
- ¿Cuáles son los costos promedios que surgen para la realización de este proceso?
- ¿Qué dispositivos para el desarrollo de aplicaciones que usen realidad virtual hay en la universidad?

Como principal aporte que trae esta investigación está la contribución para el desarrollo de aplicaciones que usen realidad virtual de una manera más fácil, y permitir la migración de un gran número de aplicaciones en 3D a realidad virtual

La tesis está estructurada en 3 capítulos los cuales son:

- Capítulo 1 Fundamentación teórica. En este capítulo se definen los elementos teóricos necesarios para el desarrollo de la investigación y los principales conceptos que se emplearán durante todo el trabajo. Se realiza un análisis de soluciones similares y se selecciona la metodología de desarrollo de software, herramientas y tecnologías utilizadas.
- Capítulo 2 Análisis y diseño. En este capítulo se describen las particularidades técnicas y los artefactos generados para el desarrollo

de la solución, como son: el diseño del componente, la especificación de los mecanismos, la descripción de los requisitos no funcionales y la arquitectura.

- Capítulo 3: Implementación y Pruebas. En este capítulo se describen las distintas tareas de ingeniería. Finalmente, se presenta el resultado de las pruebas realizadas al componente para validar su correcto funcionamiento.

Capítulo 1 Fundamentación Teórica

1.1 Introducción al Capítulo

En el presente capítulo se exponen los referentes fundamentales del trabajo investigativo. Estos buscan el fin de facilitar la comprensión del objeto de estudio, así como los análisis derivados de la investigación para una propuesta coherente. Se analizarán los principales conceptos referentes al desarrollo de la aplicación y se expondrán las diferentes técnicas y herramientas a usar a la hora de su desarrollo.

1.2 Fundamentos de la Realidad Virtual

Para una mejor comprensión de la solución. Es necesario hacer alusión a conceptos y fundamentos que llevan a comprender los sistemas que usan realidad virtual.

1.2.1 Realidad Virtual

Existen diversas definiciones del término de realidad virtual, derivadas principalmente de la combinación de tres perspectivas: la filosófica, la técnica y la psicológica [13].

Brudniy y Demilhanova definen a la realidad virtual como la forma más avanzada de relación entre una persona y un sistema informático. Dicha relación permite una interacción directa entre el usuario y el ambiente generado artificialmente, ambiente que está destinado a estimular alguno o todos los sentidos humanos, caracterizándose principalmente por crear una ilusión a nivel cerebral de participación directa en dicho ambiente [13].

Islande plantea que existen otros términos como entorno sintético, ciberespacio, realidad artificial, tecnología de simulación que han sido adoptados como sinónimos de realidad virtual [14].

Actualmente, una de las definiciones más aceptadas de la realidad virtual es simulación de un entorno real generado por un ordenador, en la que a través de una interfaz hombre-máquina se va a permitir al usuario interactuar con ciertos elementos dentro del escenario simulado [15]. En la Ilustración 2 se muestra a un usuario interactuando con un videojuego de realidad virtual.



Ilustración 2

Concretamente en este trabajo, la Realidad Virtual es un entorno de escenas y objetos de apariencia real, generado mediante tecnología informática, que crea en el usuario la sensación de estar inmerso en él. Hoy en día lo más común es que dicho entorno se contempla a través de un dispositivo conocido como gafas o casco de realidad virtual.

1.2.2 Inmersión

La inmersión en la realidad virtual es una percepción de estar físicamente presente en un mundo no físico. La percepción se crea rodeando al usuario del sistema de realidad virtual en imágenes, sonido u otros estímulos que proporcionan un entorno total fascinante [16].

Se utilizan varias técnicas físicas en el momento de dar cierto nivel de inmersión a las aplicaciones que usan Realidad Virtual. Por ejemplo, que lo que sea visto por el usuario al interactuar con la aplicación esté siendo mostrado en un par estereoscópico o rodearlo de otros dispositivos para que el entorno virtual se sienta más real. En la Ilustración 3 se muestra a un usuario jugando un videojuego de realidad virtual con el género carreras. Mueve el vehículo con un volante, acelera este con un acelerador y está sentado en una posición parecida a la que debería tener el conductor del auto. Este sistema proporciona al usuario de una alta sensación de inmersión.



Ilustración 3

1.2.3 Visión Estereoscópica

La estereoscopía es cualquier técnica capaz de recoger información visual tridimensional y/o crear la ilusión de profundidad mediante una imagen estereográfica, un estereograma o una imagen 3D [17]. La ilusión de la profundidad en una fotografía, película, u otra imagen bidimensional se crea presentando una imagen ligeramente diferente para cada ojo, como ocurre en nuestra forma habitual de ver. Muchas pantallas 3D usan este método para transmitir imágenes. Fue inventado por Sir Charles Wheatstone en 1840 [17].

En la realidad virtual la estereoscopía es usada al crear y mostrar, a partir de dos imágenes ligeramente diferentes, un par estereoscópico y reproducirlo ante el usuario para dar la sensación de profundidad y que el nivel de inmersión del usuario aumente. En la Ilustración 4 se muestra el par estereoscópico construido de una escena del juego The Legend of Zelda: Breath of the Wild.



Ilustración 4

1.3 Análisis de aplicaciones homólogas

Hay muchas aplicaciones internacionales para a partir de una imagen en 3D o 2D crear un par estereoscópico y dar una sensación de inmersión, entre ellas se encuentran:

Trinus VR

Trinus Cardboard VR proporciona a los jugadores un casco de realidad virtual de gama alta sin necesidad de comprar hardware costoso, porque si el usuario posee un móvil es suficiente. Trinus usa la pantalla y los sensores de su propio teléfono inteligente para transformarlo en un portal para sus juegos de PC [18].

Trinus VR es un sistema que está conformado por una aplicación de Android, que usa los sensores de movimiento del teléfono y los interpreta para hacer un seguimiento de los movimientos de tu cabeza y mostrar una imagen estereoscópica al usuario, y una aplicación de escritorio que creará un Punto de Acceso Wifi para hacer una captura de pantalla de una ventana en la pantalla y crear un par estereoscópico para transmitirlo a la aplicación cliente (aplicación

de Android) [19], [20]. En la Ilustración 5 se aprecia a un usuario usando la aplicación Trinus VR.



Ilustración 5

VRidge

VRidge es una tecnología que brinda a los visores de realidad virtual móviles la capacidad de ejecutar juegos de escritorio de realidad virtual. Consta de dos partes: una aplicación de escritorio y una aplicación móvil. Cuando ambos están conectados, el PC comienza a reconocer el teléfono como un visor HTC Vive [21].

La comunicación entre aplicaciones ocurre a través de un cable USB o conexión WiFi. VRidge envía la imagen del PC al teléfono como una secuencia de video de alta calidad con la latencia más baja posible. El teléfono devuelve los datos de rotación de la cabeza para permitir el movimiento de la cabeza del usuario [22]. En la Ilustración 6 se aprecia a un usuario usando la aplicación VRidge.

El resultado final es que el teléfono del usuario emparejado con el simple Google Cardboard (o cualquier otro Visor VR móvil) actúa como un visor de PC VR completamente desarrollado: admite la vista estereoscópica en 3D y los movimientos de la cabeza [22].



Ilustración 6

El software es compatible con una amplia gama de juegos, ya que es compatible con el estándar OpenVR. Todos los juegos que no dependen en gran medida de los controladores manuales serán perfectamente jugables [21].

Firefox Reality

Firefox Reality es un navegador multiplataforma de código abierto optimizado para cascos autónomos de realidad virtual, de realidad aumentada y de realidad mixta [23].

Los cascos de realidad virtual independientes, como HoloLens de Microsoft o Vive Focus, son esencialmente computadoras portátiles. Aunque está diseñado específicamente para aplicaciones independientes, Firefox Reality también puede ejecutarse en cascos que se conectan a una PC para obtener recursos y dispositivos de gama baja como visores de cartón, así como computadoras de escritorio, portátiles, tabletas y teléfonos inteligentes [24].

Es un navegador rápido y seguro que está diseñado para navegar por la web abierta en realidad virtual. Han colaborado con creadores de contenido de todo el mundo para brindar un fácil acceso a increíbles experiencias de realidad virtual que se pueden disfrutar directamente desde Firefox Reality [23].

Intugame

Intugame VR es una aplicación de juegos que te permite jugar juegos de PC existentes en realidad virtual. El juego se transmite desde la PC a su teléfono inteligente. Pones el teléfono inteligente en un casco VR y se representa en 3D estereoscópico. Los sensores del dispositivo se utilizan para seguir con precisión el movimiento de la cabeza y ajustar la cámara del juego en consecuencia. Para usar Intugame VR, se necesita un casco de realidad virtual que funcione con teléfonos inteligentes. Ya hay un montón de cascos VR en el mercado. El más accesible es Google Cardboard, pero Samsung ya lanzó una versión para consumidores de Gear VR. También necesita un teléfono inteligente Android o iOS. Los teléfonos inteligentes de gama alta son más adecuados para la aplicación, ya que hace un uso intensivo de la CPU y la GPU [25].

Virtual Desktop

Virtual Desktop es una aplicación desarrollada para los cascos Oculus Rift / Rift S, HTC Vive, Valve Index y WMR que le permite usar su computadora en VR. Esta aplicación NO es para Oculus Quest. Puede navegar por la web, ver películas, Netflix o incluso jugar juegos en una pantalla virtual gigante [26].

OpenVR Cardboard

OpenVR es una interfaz de programación de código abierto creada por Valve para permitir la comunicación con un sistema de realidad virtual. Gracias a su apertura y al apoyo de sus creadores, OpenVR es compatible con casi todos los principales cascos de gama alta del mercado, incluidos HTC Vive, Oculus Rift, cualquiera de los cascos Windows Mixed Reality, Pimax 5K/8K e incluso unos cascos para móviles como los que utilizan NoloVR y unos cascos de nueva generación como los cascos con pantalla biónica Varjo o los cascos de pequeño tamaño de Kopin [27].

El cliente de Android para OpenVRCardboard es VirtualCheap. Esta aplicación de Android está destinada a enviar datos de giroscopio a un servidor de socket para recibir datos de Android diseñado para funcionar en el lado del zócalo de la PC (servidor) Controlador OpenVR para intercambiar datos entre android y steamvr en PC, el objetivo principal es Linux [27].

1.4 Análisis comparativo de las aplicaciones homólogas

A partir del análisis realizado se ha logrado llegar a definir características comunes y diferencias entre estos sistemas determinando como características comunes:

Comunicación entre un dispositivo cliente y la PC de escritorio:

Estas aplicaciones permiten que se pueda conectar un casco de realidad virtual o un teléfono móvil. Esta conexión se establece de diferentes maneras según la aplicación en cuestión. Algunas usan un sistema de cliente-servidor con un servidor creando un servidor socket al cual se conecta un móvil con la aplicación de teléfono inteligente determinada (Ilustración 7), dando en consecuencia a un sistema que usa dos aplicaciones distintas, a este sistema también suelen integrársele un método de conexión alternativo a través de un cable USB, sin embargo este modo de conexión a través de un cable de USB trae como consecuencia una reducción de la libertad de movimiento que podría



Ilustración 7

tener el usuario al usar el sistema. Otro método es sin la necesidad del dispositivo móvil, pero esto requeriría el uso de un Oculus o casco muy específico y con un alto costo para el cliente.

Sensor para identificar la rotación de la cabeza:

Estas aplicaciones requieren que el dispositivo al que se estén comunicando tenga algún sensor que les permita hacer lecturas de movimiento para rastrear la posición de la cabeza del usuario en todo momento, y adaptar la imagen en consecuencia. La ilustración 8 muestra los principales movimientos a reconocer. En el caso de las aplicaciones que usan dispositivos móviles usan métodos para leer los cambios ocurridos en el giroscopio del teléfono y en consecuencia a estos cambios adaptar la imagen. En el caso de aplicaciones que usan cascos



Ilustración 8

de otro tipo estos están integrados con mecanismos para la lectura de estos movimientos solo se hace necesario pedir estos valores para responder en consecuencia.

Construcción del Par Estereoscópico:

La aplicación en cuestión es capaz de hacer una captura de una imagen que puede presentar gráficos 3D o 2D y con ella construir un par estereoscópico para ser reproducido al usuario en dependencia del dispositivo seleccionado, en caso de ser un móvil, se observará el par estereoscópico en pantalla y una vez el usuario se ponga el casco de realidad virtual, inserte el teléfono inteligente y ajuste las lentes, dará la sensación de estar inmerso en el entorno. En el caso

de los oculus la imagen se transmitirá a este y mostrará el par estereoscópico al usuario requiriendo que ajuste también las lentes.

1.5 Método de ingeniería de Software

Se realizó un estudio de las distintas metodologías para el desarrollo de software, y se seleccionó la metodología XP o Programación Extrema.

La metodología XP es un conjunto de técnicas que dan agilidad y flexibilidad en la gestión de proyectos. Se centra en crear un producto según los requisitos exactos del cliente. De ahí, que le involucre al máximo durante el método de gestión del desarrollo del producto. Se trata de una metodología de desarrollo cuyo objetivo es promover la aplicación de prácticas de ingeniería apropiadas para la creación de software [28].

Este tipo de metodología se impuso sobre el resto de las tecnologías estudiadas ya que pone más énfasis en la adaptabilidad que en la previsibilidad. La Programación Extrema considera que los cambios de requisitos sobre la marcha son acciones naturales e inevitables en el desarrollo de un proyecto. Creer que ser capaces de adaptarse a los cambios que puedan surgir en cualquier punto del ciclo de vida de un proyecto es una mejor previsión y más realista que intentarlos definir todos en un principio y que no varíen más.

1.6 Herramientas y Tecnologías

1.6.1 Lenguaje del Modelado

Según Pressman el Lenguaje de Modelado Unificado (UML) es “un lenguaje estándar para escribir diseños de software”, puede usarse para visualizar, especificar, construir y documentar los artefactos de un sistema de software intensivo. Este lenguaje está pensado para utilizarse en todos los métodos de desarrollo, etapas del ciclo de vida de un software, dominios de aplicación y medios. Es un sistema notacional destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. Es el estándar mundial que utilizan los desarrolladores, autores y proveedores de Herramientas para Ingeniería de Software Asistida por Computación (CASE) [29].

1.6.2 Herramienta Case

Visual Paradigm versión 8.0 es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software [30].

Las ventajas que proporciona Visual Paradigm para UML son [31]:

- Dibujo. Facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.
- Corrección sintáctica. Controla que el modelado con UML sea correcto.
- Coherencia entre diagramas. Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- Integración con otras aplicaciones. Permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad.
- Trabajo multiusuario. Permite el trabajo en grupo, proporcionando herramientas de compartición de trabajo.
- Reutilización. Facilita la reutilización, ya que disponemos de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos.
- Generación de código. Permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software.
- Generación de informes. Permite generar diversos informes a partir de la información introducida en la herramienta.

1.6.3 Unity 3D

Se usará la versión 2020.3.26f1 de esta herramienta a la hora de desarrollar el componente ya que es un potente motor multiplataforma para realizar proyectos tanto 2D como 3D. Fácil de manejar para los principiantes, permite crear fácilmente videojuegos y aplicaciones para multitud de plataformas.

Además una gran cantidad de proyectos que usan gráficos por computadora en la Universidad de Ciencias Informáticas usan esta herramienta en el desarrollo de sus proyectos y el componente está orientado a este sector [32].

1.6.4 Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el estilo del editor, los atajos de teclado y otras preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft [33].

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor utilizado en Visual Studio Team Services.

Compatibilidad: Windows, Linux, Mac OS

Complementos y funciones principales:

- Navegación sencilla para un alto rendimiento
- Admite diferentes paquetes para personalizar el editor
- Puede copiar fragmentos de código comunes con una sola palabra clave
- Cambio de proyecto instantáneo y edición dividida

Ventajas [34]:

- Visual Studio Code es una herramienta que tiene soporte nativo para gran variedad de lenguajes, entre ellos podemos destacar: C#, JavaScript, entre otros.
- Posibilidad de configurar la interfaz a nuestro gusto. De esta forma, podremos tener más de un código visible al mismo tiempo, las carpetas de nuestro proyecto y también acceso a la terminal o un detalle de problemas, entre otras posibilidades.

- Existencia de una amplísima gama de temas o estilos visuales para Visual Studio Code, que hacen el trabajo con el software más agradable a la vista.
- Goza de un soporte técnico formidable pues debido a su frecuente uso por la comunidad de desarrolladores, se puede encontrar fácilmente documentación y ayuda en foros y sitios relacionados.

1.6.5 C#

"C#" es un lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO. C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común [35].

Se planeó el uso de este lenguaje ya que es uno de los lenguajes fundamentales en los cuales se programa en la herramienta Unity 3D. Además, el equipo de trabajo está capacitado en el uso de este lenguaje para la realización rápida del proyecto.

1.7 Conclusiones del capítulo

A través de un análisis exhaustivo en el cual se tomaron en cuenta los principales conceptos relacionados con el producto a desarrollar, un estudio amplio de las aplicaciones homólogas a nuestro producto y características comunes presentes entre estos, y las herramientas más seleccionadas según la orientación de nuestro proyecto, se planeó el uso de la herramienta Unity 3D usando el lenguaje de programación C#. Concluyendo que la solución estaría conformada por dos softwares, una aplicación de escritorio y otra de móvil.

Capítulo 2 Análisis y Diseño

2.1 Introducción al capítulo

Este capítulo describe la propuesta de solución de la presente investigación. Se especifican los principales aspectos que definen el Marco de Trabajo Ingenieril para el proceso de desarrollo del componente en cuestión, así como plantear una serie de aspectos como los requisitos de la aplicación, criterios de calidad y otros.

2.2 Propuesta de Solución

La solución estará compuesta por dos softwares. Primero, una aplicación de escritorio (aplicación servidor) que podría ser integrada a los programas diseñados en Unity 3D como un paquete. El segundo, una aplicación móvil (Aplicación cliente) que una vez conectado con la aplicación de escritorio a través de la WIFI muestre el par estereoscópico, estableciendo una comunicación constante para un resultado efectivo de los procesos a realizar por ambas.

La aplicación de escritorio tendría las funciones fundamentales de crear el par estereoscópico y transmitirlo a la aplicación cliente, interpretación del gesto una vez que reciba los datos del giroscopio dados por la aplicación cliente y el entorno que es independiente de la aplicación de escritorio. El entorno depende de la aplicación en la que este paquete sea empleado.

La aplicación móvil obtendría los valores del giroscopio y los transmitiría a la aplicación de escritorio y una vez obtenido la imagen del par estereoscópico la mostraría en pantalla.

2.2.1 Diagrama de actividades

Se diseñó un diagrama de actividades, Ilustración 9, para una mejor comprensión del flujo de comunicación entre las dos aplicaciones:

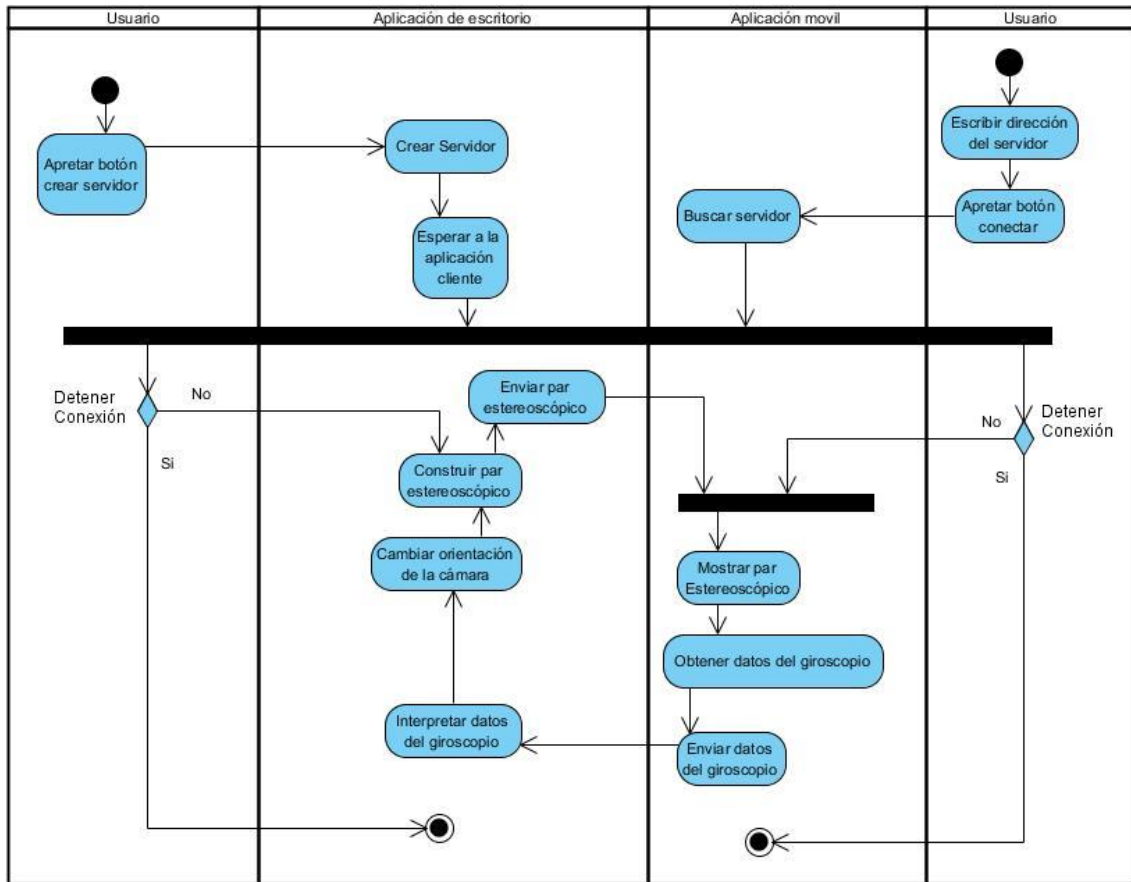


Ilustración 9

Ambos dispositivos la computadora de escritorio o laptop deben estar conectados a la misma red WIFI. Una vez iniciada la aplicación de escritorio el usuario debe de crear el servidor. Una vez iniciada la aplicación de móvil el usuario debe escribir la dirección del servidor en la red WIFI y apretar el botón de conectarse. Al encontrarse el servidor se establecerá la conexión entre ambos dispositivos, el usuario puede parar la conexión cuando lo desee finalizando el proceso. Mientras la conexión se mantenga la aplicación de escritorio construirá un par estereoscópico según la dirección de la cámara, luego enviará la imagen del par estereoscópico al móvil. Una vez recibido, la aplicación de cliente obtendrá los datos del giroscopio y los enviará al servidor. El servidor interpretara los datos del giroscopio enviados y modificara la dirección de la cámara según estos volviendo a construir al par estereoscópico para enviarlos al cliente hasta que la conexión sea detenida.

2.2.2 Requisitos Funcionales

Durante varias reuniones con el cliente se lograron determinar los distintos requisitos funcionales de la aplicación, también fue de gran ayuda el estudio de

homólogos correspondientes para determinar características comunes entre los diversos softwares que se estudiaron y tener bien definidas que características funcionales debe tener el software implementado:

Requisitos por parte de la aplicación de escritorio:

Código	Descripción
RF01	Configurar el puerto del Punto de Acceso
RF02	Generar un par estereoscópico a partir de un entorno 3D
RF03	Transmitir una imagen en streaming una vez establecida la conexión con la aplicación en el teléfono
RF04	Rotar la dirección de la cámara según los valores de los movimientos de la cabeza del usuario

Requisitos por parte de la aplicación en el teléfono:

Código	Descripción
RF05	Configurar el puerto del Punto de Acceso
RF06	Mostrar una imagen transmitida en streaming al establecer una conexión con la aplicación de escritorio
RF7	Obtener los datos del giroscopio y transmitirlos en streaming a la aplicación de escritorio

2.2.3 Requisitos no Funcionales

Tipo	Código	Descripción	
		Aplicación-Desktop	Aplicación-Móvil
Usabilidad	RNF01	Debe ser un módulo de unity capaz de una fácil integración a otros proyectos	
	RNF02	Debe ser sencillo de usar por el resto de los equipos de proyecto que lo vayan a utilizar	

	RNF03		La configuración de sus variables (puerto) deben ser intuitivas para el usuario
Seguridad	RNF04	Solo debe establecer la conexión con una única aplicación-móvil con un puerto definido.	
Diseño e Implementación	RNF05	El sistema deberá ser desarrollado en unity	
Soporte	RNF06		El sistema debe ser compatible con versiones de android v7.0 o superiores

2.2.4 Integración de ambiente RV en Unity

Para realizar la aplicación se utilizaron una serie de paquetes de la comunidad de Unity y herramientas del editor:

- XR Plug-in Magnement
- Mock HMD XR Plugin
- Multiplayer HLAPI

Durante el desarrollo de la solución, se tuvieron que hacer calibraciones a la hora de calcular los movimientos del giroscopio. Además se desarrollaron algunas funcionalidades dentro de los paquetes para la transmisión de la información entre las dos aplicaciones.

XR Plug-in Magnement

El XR Plugging Magnement es una herramienta de Unity 3D que se encuentra en versiones posteriores a la 2019.x. Si se planea su uso en algún proyecto deberán bajarse una serie de librerías, proceso que se puede hacer permitiéndole acceso a internet a Unity.

La herramienta posee un conjunto de librerías para el trabajo con la realidad extendida, XR en sus siglas en ingles. La realidad extendida es un concepto que engloba la realidad aumentada y realidad virtual.

Para usar esta herramienta en un proyecto de unity se debe ir a la pestaña Project Setting -> XR Plug-in Management. Se encuentran una serie de opciones distintas para el uso de realidad virtual usando distintos dispositivos, en este caso se usó el Mock HMD Loader, que es el que se ajusta a las necesidades de la aplicación.

Mock HMD XR Plugin

Mock HMD XR Plugin permite construir un par estereoscópico desde la computadora sin tener que estar trabajando con un casco de realidad virtual. Para trabajar con él, primero ha de ser seleccionado dentro la pestaña de Unity Project Settings -> XR Plug-in Management y seleccionarlo según la plataforma en la que se esté trabajando.

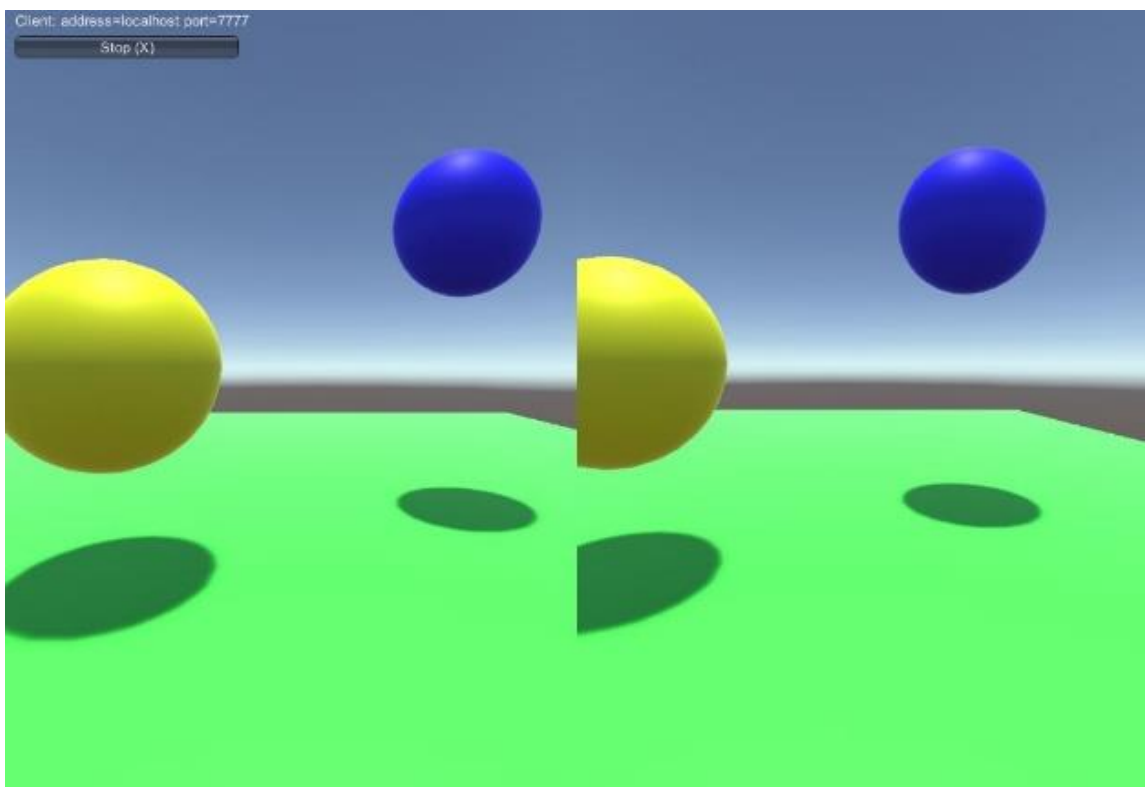


Ilustración 10

La Ilustración 10 muestra como se ve la construcción del par estereoscópico desde la aplicación cliente.

Para que empiece a construir el par estereoscópico y mostrarlo se debe de primero configurar el entorno para que se ejecute manualmente y luego iniciar la construcción del par estereoscópico haciendo que la aplicación pase a modo de realidad virtual. Las siguientes líneas de código permiten que este proceso sea realizado:

```
UnityEngine.XR.Management.XRGeneralSettings.Instance.Manager.InitializeLoaderSync();  
UnityEngine.XR.Management.XRGeneralSettings.Instance.Manager.StartSubsystems();
```

Para detener el modo de realidad virtual solo hay que llamar al método StopSubsystems(). Esto permite que puedas volver a cambiar al modo de realidad virtual llamando a StartSubsystems() en cualquier momento:

```
UnityEngine.XR.Management.XRGeneralSettings.Instance.Manager.StopSubsystems();
```

Multiplayer HLAPI

Multiplayer HLAPI se usó para establecer una conexión entre la aplicación servidor y la aplicación cliente. Este paquete viene integrado con una interfaz denominada NetworkManagerHUD. Este script se debe añadir como un

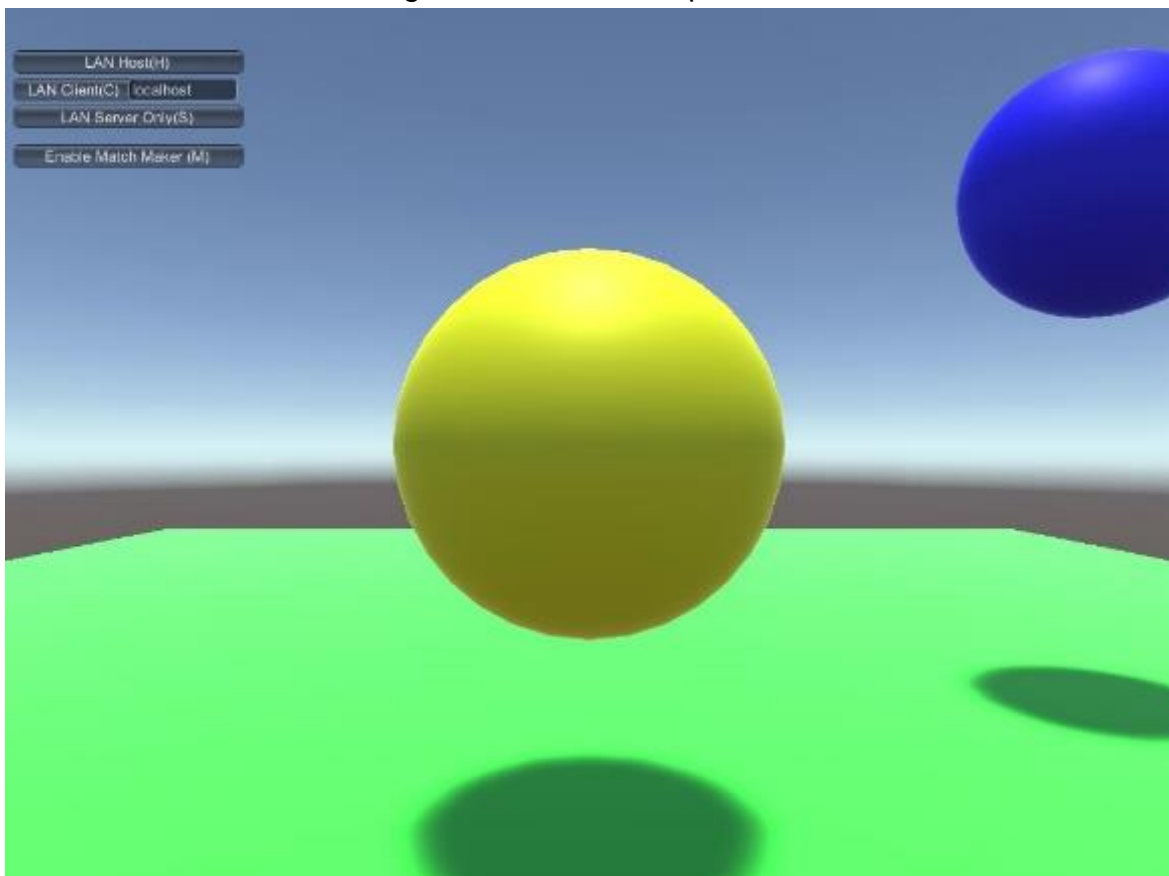


Ilustración 11

componente a algún objeto en la escena y si se desea que el usuario no vea este componente se debe cambiar el atributo showGUI a false. La ilustración 11 muestra la interfaz del NetworkManagerHUD en la aplicación.

Se añadió también a un componente en la escena, el componente NetworkManager que es la clase principal sobre la que gira todo el paquete. A través de él se puede configurar el puerto de conexión, establecer la cantidad de canales de comunicación, determinar si el servidor debe estar vinculado a una dirección determinada en la red. Se configuró para que el máximo de conexiones fuera de una, ya que solo puede haber un dispositivo cliente conectado, y se establecieron dos canales de comunicación para recibir y enviar el par estereoscópico y otro para el giroscopio.

Para enviar la información del giroscopio se utilizaría el canal con id = 0 y para la transmisión de la imagen se usara el que tiene id = 1. En la aplicación de escritorio se creó una clase para que dentro de su método Update() llamó al método Send() pasándole como parámetros el identificador del canal y el monitor en el cual se está calculando el par estereoscópico. También se creó otra clase con un atributo NetworkEventType y dentro de su método Update() se llamó al método Recieve() que guardará los datos y se reproducirá el monitor o display. Estas clases se añadieron como componentes en el mismo objeto que posea el NetworkManager. Se usó una analogía similar en la aplicación cliente. En este caso en vez de enviar el display se envió el cuaternio del giroscopio.

Calibración del Giroscopio

Para trabajar con el giroscopio en una aplicación móvil se debe primero habilitarlo. Todas las aplicaciones para ahorrar batería inician con el giroscopio desactivado. Esto se hace de la siguiente manera:

```
if (SystemInfo.supportsGyroscope)
{
    Input.gyro.enabled = true;
}
```

Luego de esto se debe obtener los datos del giroscopio, los cuales son devueltos en un cuaternio. El cuaternio esta representado de esta manera $A+Bi+Cj+Dk$; A, B, C y D son números reales y i, j, k son los cuaternion básicos.

Para la correcta orientación del giroscopio el objeto a rotar debe ser orientado en una dirección específica usando:

```
transform.rotation = Quaternion.Euler(90f, -90f, 0f);
```

La información de la rotación del giroscopio es muy compleja y recoge muchos datos al ser devuelta en un cuaternio. Para calibrarlo para que solo reconozca los movimientos del cuello se debe de crear otro cuaternio con los valores (0,0,1,0) y multiplicarlo por el devuelto por el giroscopio esto hará que solo interprete los valores de j que son los movimientos que estamos esperando reconocer.

Además hay que tomar en cuenta que el giroscopio reaccionara al mínimo movimiento del móvil, resultando en que sea muy difícil controlar la dirección a la que se está intentando mirar. Esto se resuelve depreciando los valores que estén por debajo de cero en el cálculo de rotación del cuaternio:

```
Quaternion rotMin = Quaternion.Euler(new Vector3(0, 0, 0));  
Quaternion rotation = transform.rotation;
```

```
if (rotation.y < rotMin.y)  
{  
    transform.eulerAngles = Vector3.zero;  
}
```

2.3 Historias de Usuario

HISTORIA DE USUARIO	
Número: 1	Usuario: Especialista
Nombre: Determinar el puerto	
Prioridad: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración: 1
Descripción: el usuario debe determinar el puerto sobre el que se establecerá la comunicación	

HISTORIA DE USUARIO	
Número: 2	Usuario: Especialista
Nombre: Construcción del par estereoscópico	
Prioridad: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración: 1
Descripción: se debe construir un par estereoscópico con los datos brindados por la cámara	

HISTORIA DE USUARIO	
Número: 3	Usuario: Especialista
Nombre: Transmisión de la imagen	
Prioridad: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración: 2
Descripción: La aplicación de escritorio debe transmitir al dispositivo móvil la imagen del par estereoscópico	

HISTORIA DE USUARIO	
Número: 4	Usuario: Especialista
Nombre: Configuración de la imagen	
Prioridad: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración: 3

Descripción: los valores de rotación de la cámara deben cambiar para coincidir con la posición en la que el usuario este mirando

HISTORIA DE USUARIO	
Número: 5	Usuario: Especialista
Nombre: Configurar el puerto	
Prioridad: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración: 2
Descripción: la aplicación de móvil debe configurar su punto de acceso para coincidir con la del servidor y establecer la comunicación	

HISTORIA DE USUARIO	
Número: 6	Usuario: Especialista
Nombre: Mostrar el par estereoscópico	
Prioridad: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración: 2
Descripción: mostrar un par estereoscópico en la pantalla de la aplicación móvil	

HISTORIA DE USUARIO	
Número: 7	Usuario: Especialista
Nombre: Transmitir los datos del giroscopio	

Prioridad: Alta	Riesgo en Desarrollo: Alta
Puntos Estimados: 2	Iteración: 3
Descripción: se deben transmitir los datos del giroscopio a la aplicación de escritorio	

2.4 Plan de Iteraciones

En la siguiente tabla se muestran del plan de iteraciones según las historias de usuario planteadas:

Iteración	Historia de Usuario	Duración
1	Determinar el puerto	2
1	Construcción del par estereoscópico	2
2	Configurar el puerto	2
2	Transmisión de la imagen	2
2	Mostrar el par estereoscópico	2
3	Transmitir los datos del giroscopio	2
3	Configuración de la imagen	2
Total		14

2.5 Plan de Entrega

En la siguiente tabla se muestran del plan de entrega según las historias de usuario planteadas:

Historia de Usuario	Puntos estimados
Determinar el puerto	2
Construcción del par estereoscópico	2
Configurar el puerto	2
Transmisión de la imagen	2
Mostrar el par estereoscópico	2
Transmitir los datos del giroscopio	2
Configuración de la imagen	2

Total	14
-------	----

2.6 Tarjetas Clase-Responsabilidad-Colaboración

En la metodología XP para el diseño de software orientado a objetos son usadas comúnmente las tarjetas Clase-Responsabilidad-Colaboración (CRC). Las Historias de Usuario son evaluadas y divididas en tareas, cada una de las cuales representa una característica del sistema. Los contenidos fundamentales de estas tarjetas son el nombre de la clase en cuestión, las responsabilidades, métodos y objetos principales de la clase, y las colaboraciones que no son más que las clases que las ayudan a cumplir estas responsabilidades.

TARJETA CRC	
Clase: CAMARA_SRIPT	
Responsabilidad	Colaboración
<ol style="list-style-type: none"> 1. Un script que va a poder ser añadido como un componente un objeto del tipo cámara dentro del Editor Unity, y que permitirá el trabajo del resto de los componentes de la aplicación a los datos y configuraciones del objeto cámara en cuestión 2. Se encargará del movimiento de la cámara según los datos de DESKTOP_RECIVER 	Cameraltem

TARJETA CRC	
Clase: STEREOSCOPIC_MAKER	
Responsabilidad	Colaboración
<ol style="list-style-type: none"> 1. A partir de una imagen obtenida creara una imagen con un par estereoscópico, según las indicaciones y atributos que se encuentren en el archivo STEREOSCOPIC_CONFIG 	CAMARA_SCRIPT STEREOSCOPIC_CONFIG

TARJETA CRC	
Clase: STEREOSCOPIC_CONFIG	
Responsabilidad	Colaboración

1. Contendrá una serie de atributos como la distancia interpupilar y otros que serán usadas en la creación del par estereoscópico	
---	--

TARJETA CRC	
Clase: AdHoc_MAKER	
Responsabilidad	Colaboración
1. A partir de una serie de atributos ubicados en el archivo AdHoc_CONFIG creara un punto de acceso para que establecer la comunicación entre ambas aplicaciones	AdHoc_CONFIG

TARJETA CRC	
Clase: AdHoc_CONFIG	
Responsabilidad	Colaboración
1. Contendrá una serie de atributos necesarios en la creación del punto de acceso como son: nombre del punto de acceso, contraseña del punto de acceso, y puerto a través del cual se comunicarán ambas aplicaciones	

TARJETA CRC	
Clase: DESKTOP_SENDER	
Responsabilidad	Colaboración
1. Enviará la imagen del par estereoscópico a el dispositivo conectado usando el protocolo de comunicación UDP a la aplicación móvil conectada a el punto de acceso	AdHoc_MAKER STEREOSCOPIE_MAKER APK_RECIVER

TARJETA CRC	
Clase: DESKTOP_RECIVER	
Responsabilidad	Colaboración
1. Recibirá los valores transmitidos por la aplicación móvil para ser usados por los	AdHoc_MAKER APK_SENDER

componentes de la aplicación de escritorio	
--	--

TARJETA CRC	
Clase: APK_CONNECTION_VERIFIER	
Responsabilidad	Colaboración
1. Guardará la información del puerto a través del cual se establecerá la transmisión con la aplicación de escritorio y verificará que se haya establecido una conexión entre ambas aplicaciones	AdHoc_MAKER

TARJETA CRC	
Clase: APK_RECIVER	
Responsabilidad	Colaboración
1. Recibirá los valores transmitidos por la aplicación de escritorio para ser usados por los componentes de la aplicación de móvil, estos componentes serán una imagen con un par estereoscópico	AdHoc_MAKER APK_CONNECTION_VERIFIER

TARJETA CRC	
Clase: APK_SCREEN	
Responsabilidad	Colaboración
1. Mostrará la imagen del par estereoscópico en la pantalla del teléfono móvil	APK_RECIVER

TARJETA CRC	
Clase: APK_GIROSCOPIO	
Responsabilidad	Colaboración
1. Obtendrá constantemente los valores del giroscopio del dispositivo móvil	

TARJETA CRC	
Clase: APK_SENDER	
Responsabilidad	Colaboración

1. Obtendrá los valores que se encuentran en APK_GIROSCOPIO y los transmitirá a la aplicación de escritorio usando el protocolo de comunicación UDP	APK_GIROSCOPIO
---	----------------

2.8 Estilo Arquitectónico

La arquitectura de software de un sistema representa las decisiones de diseño relacionadas con la estructura y el comportamiento general del sistema. La arquitectura ayuda a las partes interesadas a comprender y analizar cómo el sistema logrará cualidades esenciales como la adaptabilidad, la disponibilidad y la seguridad. [36]

La arquitectura de software admite el análisis de las cualidades del sistema cuando los equipos toman decisiones sobre el sistema en lugar de después de la integración o implementación. Ya sea que esté diseñando un nuevo sistema, desarrollando un sistema exitoso o modernizando un sistema heredado, este análisis oportuno permite a los equipos determinar si los enfoques que han elegido generarán una solución aceptable. Una arquitectura efectiva sirve como el pegamento conceptual que mantiene unida cada fase del proyecto para todas las partes interesadas, lo que permite la agilidad, el ahorro de tiempo y costos, y la identificación temprana de los riesgos de diseño. [36]

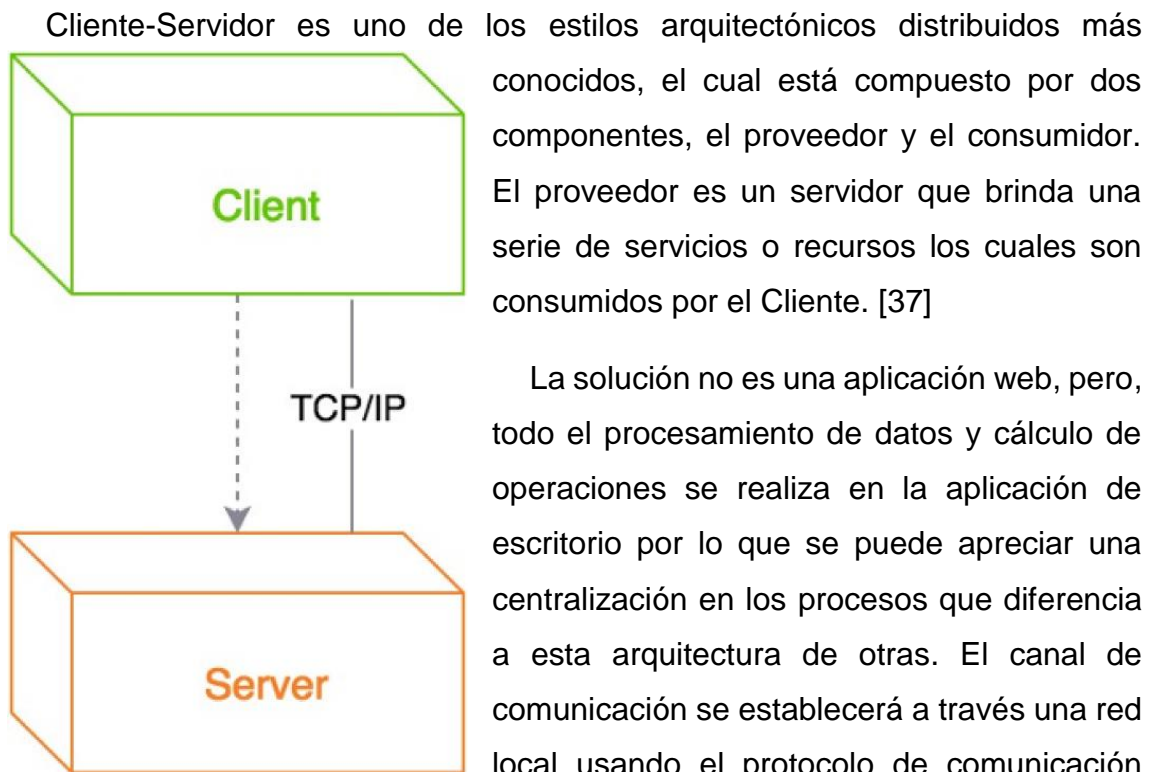


Ilustración 12

TCP/IP.

2.10 Conclusiones del capítulo

En este capítulo se analizaron los elementos del diseño de la aplicación, plasmando los requisitos funcionales y no funcionales del sistema, su descripción en las historias de usuario, la utilización del método de las tarjetas CRC y se explicó el cómo se desarrollaría la propuesta de solución.

Capítulo 3 Implementación y Pruebas

3.1 Introducción al capítulo

Este capítulo está centrado en la descripción de la implementación de la solución, así como las pruebas de aceptación que se le hicieron al software. Se inicia con las tareas de ingeniería para las cuales se tuvo en cuenta el punto de vista del programador. Se realiza una descripción de la misma. En la culminación del capítulo se describirán las distintas pruebas de aceptación realizadas sobre el software.

3.2 Tareas de Ingeniería

3.2.1 Iteración 1

En la primera iteración se concentra en la aplicación de escritorio para dar pie al desarrollo conjunto de ambas aplicaciones en la siguiente iteración:

1. Mostrar una interfaz que permita establecer un puerto para la comunicación
2. Construir un par estereoscópico

Tarea	
Número de Tarea 1	Historia de usuario:1
Nombre: Mostrar una interfaz que permita establecer un puerto para la comunicación.	
Tipo de Tarea: Desarrollo	Puntos estimados: 2
Programador: Lester Atreyu Reyes Capote	
Descripción: Se incorporaron las clases NetworkManager y NetworkManagerUI configurando los valores donde se desee que aparezca la interfaz	

Tarea	
Número de Tarea 2	Historia de usuario:2
Nombre: Construir un par estereoscópico	
Tipo de Tarea: Desarrollo	Puntos estimados: 2
Programador: Lester Atreyu Reyes Capote	

Descripción: Se creó un nuevo objeto en escena y añadirle el componente XR Rig y establecer las configuraciones deseadas en player setting

3.2.2 Iteración 2

En esta iteración se encuentran las tareas fundamentales que permitirán la comunicación y demostrarán que esta está sucediendo.

3. Mostrar una interfaz que permita cambiar el puerto para que sea igual que la del puerto del servidor.
4. La aplicación de escritorio debe transmitir la imagen del par estereoscópico a la aplicación cliente
5. La aplicación de cliente debe mostrar un par estereoscópico en la pantalla

Tarea	
Número de Tarea 3	Historia de usuario:3
Nombre: Mostrar una interfaz que permita cambiar el puerto para que sea igual que la del puerto del servidor	
Tipo de Tarea: Desarrollo	Puntos estimados: 2
Programador: Lester Atreyu Reyes Capote	
Descripción: Se incorporaron las clases NetworkManager y NetworkManagerUI configurando los valores donde se desee que aparezca la interfaz y asegurarse que coincida con el puerto del servidor	

Tarea	
Número de Tarea 4	Historia de usuario:5
Nombre: La aplicación de escritorio debe transmitir la imagen del par estereoscópico a la aplicación cliente	
Tipo de Tarea: Desarrollo	Puntos estimados: 2
Programador: Lester Atreyu Reyes Capote	
Descripción: Se usaron los métodos del paquete UNET y el método NetworkConnection.Send() enviando los datos de la imagen del par estereoscópico	

Tarea	
Número de Tarea 5	Historia de usuario:6
Nombre: La aplicación de cliente debe mostrar un par estereoscópico en la pantalla	
Tipo de Tarea: Desarrollo	Puntos estimados: 2
Programador: Lester Atreyu Reyes Capote	
Descripción: Hacer que la imagen sea mostrada en la pantalla del móvil cambiando los valores en las pestañas de configuración del proyecto	

3.2.3 Iteración 3

En esta iteración se realizaron las tareas finales para que permitirán la transmisión de los datos desde la aplicación cliente a la aplicación de escritorio.

6. Transmitir los datos del giroscopio desde la aplicación cliente a la aplicación de escritorio.
7. La aplicación de escritorio debe modificar los valores de rotación de la cámara según los valores dados

Tarea	
Número de Tarea 6	Historia de usuario:7
Nombre: Transmitir los datos del giroscopio desde la aplicación cliente a la aplicación de escritorio.	
Tipo de Tarea: Desarrollo	Puntos estimados: 2
Programador: Lester Atreyu Reyes Capote	
Descripción: Se usaron los métodos del paquete UNET y el método NetworkConnection.Send() enviando los datos del giroscopio	

Tarea	
Número de Tarea 7	Historia de usuario:4
Nombre: La aplicación de escritorio debe modificar los valores de rotación de la cámara según los valores dados	
Tipo de Tarea: Desarrollo	Puntos estimados: 2
Programador: Lester Atreyu Reyes Capote	

Descripción: Incluir un script a la cámara que implemente el método transform.rotation que se le pasaran los valores del giroscopio

3.3 Descripción de la Solución

La solución es un sistema-demo compuesto por dos aplicaciones. En la aplicación de escritorio se encuentra una escena con dos esferas y una interfaz que permite la creación de un servidor. En la aplicación de móvil se encuentra una escena con una interfaz que permite conectarse al servidor al ingresar el puerto y dirección del servidor.

Una vez establecida la conexión se mostrará en la pantalla de la aplicación cliente el par estereoscópico en el cual habrá dos imágenes ligeramente diferentes y en según la rotación del teléfono se verá el resto de la escena en ambas aplicaciones.

3.4 Pruebas de Aceptación

Las prueba de aceptación son una técnica de testeo realizada para determinar si el sistema de software ha cumplido o no con las especificaciones de requisitos. El objetivo principal de esta prueba es evaluar el cumplimiento del sistema con los requisitos del negocio y verificar si cumple con los criterios requeridos para la entrega a los usuarios finales. [38]

3.4.1 Iteración 1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Determinar el puerto	
Descripción: Prueba para comprobar que se puede introducir un puerto para establecer la comunicación	
Condiciones de ejecución: El usuario debe poder elegir a través de qué número de puerto se establecerá la comunicación	
Pasos de ejecución: El usuario debe introducir un número de puerto valido entre el 0 al 65535 y apretar el botón LAN Host()	
Resultados esperados: De ser correcto los datos se mostrará una pestaña con las etiquetas servidor y el cliente	

Caso de prueba de aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Construcción del par estereoscópico	
Descripción: Prueba para comprobar que se está creando el par estereoscópico.	
Condiciones de ejecución: El usuario debe poder visualizar el par estereoscópico generado de la imagen	
Pasos de ejecución: Una vez iniciada la aplicación en el usuario debe poder apretar un botón que le permita cambiar de una imagen a la otra	
Resultados esperados: De exitoso mostrará el par estereoscópico o la imagen sin el par estereoscópico	

3.4.2 Iteración 2

Caso de prueba de aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Configurar el puerto	
Descripción: Prueba para comprobar que se puede cambiar el número de puerto de comunicación en la aplicación cliente	
Condiciones de ejecución: El usuario debe poder cambiar el número de puerto para hacerlo coincidir con el de la aplicación de escritorio	
Pasos de ejecución: El usuario debe introducir un número de puerto válido entre el 0 al 65535 y apretar el botón LAN Client()	
Resultados esperados: De ser correcto los datos se mostrará una pestaña con el servidor y el cliente	

Caso de prueba de aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Transmisión de la imagen	
Descripción: Prueba para comprobar que el servidor está enviando los datos la imagen a la aplicación cliente	
Condiciones de ejecución: El usuario debe comprobar que se establece la transmisión de la imagen la cual será visualizada en la pantalla	

Pasos de ejecución: Una vez apretado en el botón LAN Client() si la dirección del servidor era correcta se visualizará el par estereoscópico
Resultados esperados: De ser correcto los datos se mostrará en la pantalla el par estereoscópico

Caso de prueba de aceptación	
Código: HU6_P1	Historia de Usuario: 6
Nombre: Mostrar el par estereoscópico	
Descripción: Prueba para comprobar que se muestra la imagen del par estereoscópico en la pantalla de la aplicación cliente	
Condiciones de ejecución: El usuario debe poder visualizar el par estereoscópico en la pantalla del teléfono	
Pasos de ejecución: Una vez apretado en el botón LAN Client() si la dirección del servidor era correcta se visualizara el par estereoscópico	
Resultados esperados: De ser correcto los datos se mostrará en la pantalla el par estereoscópico	

3.4.3 Iteración 3

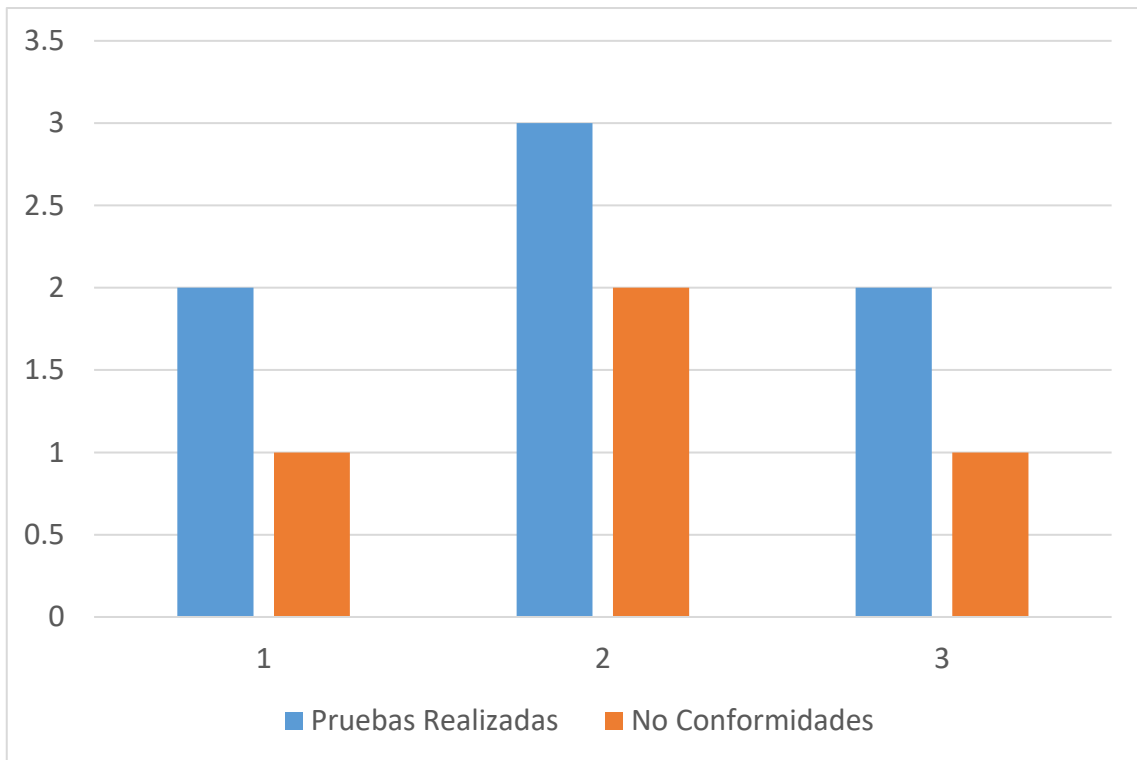
Caso de prueba de aceptación	
Código: HU7_P1	Historia de Usuario: 7
Nombre: Transmitir los datos del giroscopio	
Descripción: Prueba para comprobar que la aplicación cliente está enviando los datos del giroscopio a la aplicación servidor	
Condiciones de ejecución: El usuario debe poder ver que en la pantalla de la aplicación de escritorio que según los movimientos del teléfono la imagen mostrada cambia	
Pasos de ejecución: Una vez que la aplicación cliente establece la conexión transmitirá constantemente los datos del giroscopio	
Resultados esperados: De ser correcto los datos se mostrará que la imagen de la pantalla de la aplicación de escritorio cambia	

Caso de prueba de aceptación

Código: HU4_P1	Historia de Usuario: 4
Nombre: Configuración de la imagen	
Descripción: Prueba para comprobar que se rota la cámara en el sentido en el que se esté rotando el teléfono	
Condiciones de ejecución: El usuario debe poder ver que en la pantalla de la aplicación de escritorio que según los movimientos del teléfono la imagen mostrada cambia	
Pasos de ejecución: Una vez que la aplicación cliente establece la conexión transmitirá constantemente los datos del giroscopio	
Resultados esperados: De ser correcto los datos se mostrará que la imagen de la pantalla de la aplicación de escritorio cambia	

3.3.4 Resultados de las pruebas

Para la realización de las pruebas se dividieron en tres iteraciones fundamentales. En la primera iteración se encontró que hubo una no conformidad referida a la construcción del par estereoscópico que no se realizaba de la manera deseada, esta fue solucionada eficazmente; en la segunda iteración hubo un total de dos no conformidades en este caso orientadas a problemas en la transmisión de la imagen por la velocidad de comunicación, y la otra referida a la hora de mostrar el par estereoscópico, en el que no se ajustaba a la resolución del dispositivo móvil en cuestión, estos problemas fueron resueltos. En la tercera iteración se halló una no conformidad referida a la configuración de la imagen porque los valores del giroscopio afectaban la rotación de la cámara y no coincidía la dirección de la cámara con la dirección en la que miraba el usuario, esto fue calibrado solucionando el problema. Como se muestra en la siguiente gráfica.



3.4 Conclusiones del capítulo

En este capítulo se analizaron las distintas tareas de ingeniería de la aplicación según cada una de las iteraciones y según los métodos para la realización de las pruebas de la metodología programación extrema se mostraron las pruebas realizadas sobre el software para su aceptación final.

Conclusiones generales

Con este trabajo se desarrolló un componente que permite la aplicación de tecnologías de realidad virtual a los entornos en 3D de los distintos videojuegos en la Universidad de Ciencias Informáticas, arribando a las siguientes conclusiones:

- El desarrollo del componente permite demostrar cómo es posible proveerle a aplicaciones con entornos 3D en Unity, la capacidad para que puedan usar la Realidad Virtual.
- Es posible reducir la latencia de la transmisión de las dos aplicaciones según la renderización de la imagen desde la parte del servidor.
- Al establecer que la aplicación servidor realizaría todas las tareas de cómputo del renderizado de la imagen, cálculo del par estereoscópico y calibración del giroscopio se redujo la carga de procesos en la aplicación cliente observando un mejor rendimiento.
- Se determinó que los procesos más complejos en la realización del software se encuentran en los procesos de la calibración del giroscopio, la transmisión y renderizado de la imagen.

Recomendaciones

Con los resultados obtenidos en la investigación, diseño, implementación y prueba de la solución; se puede recomendar extender estas herramientas de la aplicación a ser probadas con otros dispositivos de realidad virtual que no se limiten a las VR Cardboard, y que se implemente también su uso para ser compatible con distintos Oculus del mercado.

Referencias

[1] T. Noordeyana, S. Mawar, y R. Emy Azziaty, «Post-Covid Tourisms: Virtual Tourisms As The New Travel Trends - ProQuest», 1 de junio de 2022. <https://www.proquest.com/openview/02c24572b31093ccdfd84d5ddf3f5d7b/1?p-q-origsite=gscholar&cbl=616374> (accedido 21 de noviembre de 2022).

[2] L. Hyunae, J. Timothy Hyungsoo, M. C. Tom Dieck, y N. Chung, «Experiencing immersive virtual reality in museums», *Inf. Manage.*, vol. 57, n.º 5, p. 103229, jul. 2020, doi: 10.1016/j.im.2019.103229.

[3] A. I. Corregidor-Sánchez, A. Segura-Fragoso, M. Rodríguez-Hernández, C. Jiménez-Rojas, B. Polonio-López, y J. J. Criado-Álvarez, «Effectiveness of virtual reality technology on functional mobility of older adults: systematic review and meta-analysis», *Age Ageing*, vol. 50, n.º 2, pp. 370-379, feb. 2021, doi: 10.1093/ageing/afaa197.

[4] K. Ahir, K. Govani, R. Gajera, y M. Shah, «Application on Virtual Reality for Enhanced Education Learning, Military Training and Sports», *Augment. Hum. Res.*, vol. 5, n.º 1, p. 7, nov. 2019, doi: 10.1007/s41133-019-0025-2.

[5] I. CORPORATIVA, «Virtual Reality: another world within sight», *Iberdrola*, 31 de enero de 2019. <https://www.iberdrola.com/innovation/virtual-reality> (accedido 21 de noviembre de 2022).

[6] L. G. V. Caraballosa y L. A. Hernández, «Experiencias en la creación de ambientes virtuales de aprendizaje desde un proyecto productivo del Polo de Realidad Virtual en la Universidad de las Ciencias Informáticas», p. 16.

[7] A. Véliz Vega, O. C. Madrigal, V. Kugurakova, A. Véliz Vega, O. C. Madrigal, y V. Kugurakova, «Aprendizaje adaptativo basado en Simuladores de Realidad Virtual», *Rev. Cuba. Cienc. Informáticas*, vol. 15, n.º 2, pp. 138-157, jun. 2021.

[8] A. H. Paez y P. M. S. Leyva, «Procedimiento de Arquitectura de Información para los entornos de Realidad Virtual con fines educativos», *Ser. Científica Univ. Las Cienc. Informáticas*, vol. 9, n.º 4, pp. 1-14, 2016.

[9] «Resin Lenses Cardboard 3D VR Virtual Reality Glasses,Movie Video Game VR Headset Black&21 * 8 * 9cm Comfortable and Environmentally: Amazon.ca: Electronics». https://www.amazon.ca/Cardboard-Virtual-Reality-Comfortable-Environmentally/dp/B09P1CK17C/ref=sr_1_27?keywords=google+Cardboard&qid=1669086801&qu=eyJxc2MiOil0LjE4liwicXNhIjoiMy4wMSIsInFzcCI6IjluODAlfQ%3D%3D&sr=8-27 (accedido 21 de noviembre de 2022).

[10] «Jojomino 2X for Google Cardboard Virtual Reality VR BiConvex Lenses Only 37mm x 45mm: Amazon.ca: Electronics». https://www.amazon.ca/Jojomino-Cardboard-Virtual-Reality-BiConvex/dp/B0BG8NS246/ref=sr_1_30?keywords=google+Cardboard&qid=1669086801&qu=eyJxc2MiOil0LjE4liwicXNhIjoiMy4wMSIsInFzcCI6IjluODAlfQ%3D%3D&sr=8-30 (accedido 21 de noviembre de 2022).

[11] «Trinus VR», *Trinus Virtual Reality*. <https://trinusvirtualreality.com/trinus-dream/> (accedido 22 de noviembre de 2022).

[12]W. Auccahuasi *et al.*, «Design of a low-cost virtual reality system, as an aid mechanism in online classes for children», p. 7.

[13]A. Brudniy y A. Demilhanova, «The Virtual Reality in a Context of the “Mirror Stage”», p. 5.

[14]Daniel Fällman, Anders Backman, y Kenneth Holmlund, «VR in Education: An Introduction to Multisensory Constructivist Learning Environments», p. 10, 1999.

[15]S. Viñas-Diz y M. Sobrido-Prieto, «Realidad virtual con fines terapéuticos en pacientes con ictus: revisión sistemática», *Neurología*, vol. 31, n.º 4, pp. 255-277, may 2016, doi: 10.1016/j.nrl.2015.06.012.

[16]R. Carter, «What is Immersive Media: An Introduction», *XR Today*, 10 de mayo de 2021. <https://www.xrtoday.com/mixed-reality/what-is-immersive-media-an-introduction/> (accedido 22 de noviembre de 2022).

[17]M. Nascimento, S. Gallo, V. Ferro, A. Brianza, E. González, y L. Vázquez, «El desarrollo del lenguaje audiovisual específico de la técnica 3D estereoscópica en la era digital», *Anu. Investig. USAL*, n.º 4, Art. n.º 4, mar. 2018,

Accedido: 22 de noviembre de 2022. [En línea]. Disponible en: <https://p3.usal.edu.ar/index.php/anuarioinvestigacion/article/view/4222>

[18] «Trinus Cardboard VR», *Trinus Virtual Reality*. <https://trinusvirtualreality.com/trinus-cardboard/> (accedido 22 de noviembre de 2022).

[19] «Trinus CBVR - Apps on Google Play». https://play.google.com/store/apps/details?id=com.loxai.trinus.full&hl=en_US&gl=US (accedido 22 de noviembre de 2022).

[20] «Trinus VR Server», *Softonic*. <https://trinus-vr-server.en.softonic.com> (accedido 22 de noviembre de 2022).

[21] «VRidge - Play PC VR on your Cardboard». <https://riftcat.com//vridge> (accedido 22 de noviembre de 2022).

[22] «VRidge FAQ», *RiftCat - Help Center*. <https://support.riftcat.com/hc/en-us/articles/360000146105-VRidge-FAQ> (accedido 22 de noviembre de 2022).

[23] «Mozilla Labs || Firefox Reality». <https://labs.mozilla.org/projects/firefox-reality/> (accedido 22 de noviembre de 2022).

[24] «Firefox Reality for VR headsets | Firefox Reality Help». <https://support.mozilla.org/en-US/kb/firefox-reality-vr-headsets> (accedido 22 de noviembre de 2022).

[25] «What is Intugame VR?» <https://www.quarkxr.com/post/get-intugame-vr> (accedido 22 de noviembre de 2022).

[26] «What are Virtual Desktops? | VMware Glossary», *VMware*. <https://www.vmware.com/topics/glossary/content/virtual-desktops.html> (accedido 22 de noviembre de 2022).

[27] «API Documentation - ValveSoftware/openvr Wiki», *GitHub*. <https://github.com/ValveSoftware/openvr> (accedido 22 de noviembre de 2022).

[28] «Metodología XP o Programación Extrema: ¿Qué es y cómo aplicarla?», *Gestor de proyectos online*. <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp> (accedido 22 de noviembre de 2022).

[29] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. Palgrave Macmillan, 2005.

[30] «Visual Paradigm Product Overview». https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html (accedido 22 de noviembre de 2022).

[31] «Ventajas de Visual Paradigm | PDF», *Scribd*. <https://es.scribd.com/document/435649983/Ventajas-de-visual-paradigm-docx> (accedido 22 de noviembre de 2022).

[32] I. Zamojc, «Introduction to Unity3D», *Code Envato Tuts+*. <https://code.tutsplus.com/tutorials/introduction-to-unity3d--mobile-10752> (accedido 22 de noviembre de 2022).

[33] «Documentation for Visual Studio Code». <https://code.visualstudio.com/docs> (accedido 22 de noviembre de 2022).

[34] «Why Visual Studio Code?». <https://code.visualstudio.com/docs/editor/whyvscode> (accedido 22 de noviembre de 2022).

[35] Bill Wagner, «A tour of C# - Overview». <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> (accedido 22 de noviembre de 2022).

[36] «Software Architecture | Software Engineering Institute». <https://www.sei.cmu.edu/our-work/software-architecture/index.cfm> (accedido 2 de diciembre de 2022).

[37] O. Blancarte, «Arquitectura Cliente-Servidor». <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor> (accedido 2 de diciembre de 2022).

[38] «Acceptance Testing». https://www.tutorialspoint.com/software_testing_dictionary/acceptance_testing.htm (accedido 2 de diciembre de 2022).