

**Universidad de las Ciencias Informáticas**

**Facultad 4**



**Título: Módulo para visualizar el Paseo Virtual UCI con el visor  
Cardboard**

**Trabajo de diploma para optar por el título de ingeniero de ciencias  
informáticas**

**Autor:** Mario Jorge Ramírez Pérez

**Tutora:** Liudmila Pupo Peña

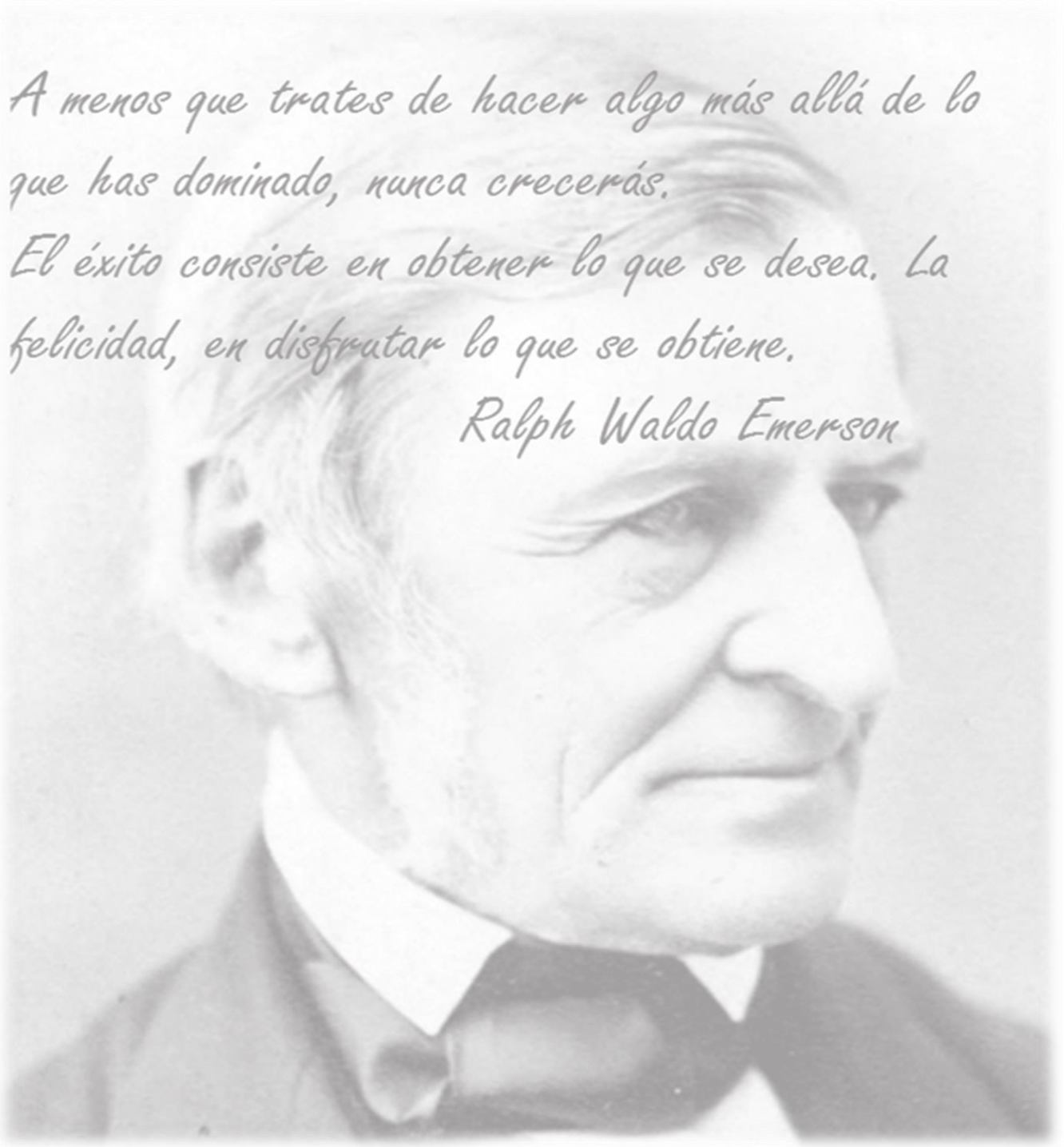
Junio de 2018

## Frase

*A menos que trates de hacer algo más allá de lo que has dominado, nunca crecerás.*

*El éxito consiste en obtener lo que se desea. La felicidad, en disfrutar lo que se obtiene.*

*Ralph Waldo Emerson*



## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis que tiene por título: “Módulo para visualizar el Paseo Virtual UCI con el visor Cardboard” y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Mario Jorge Ramírez Pérez**

\_\_\_\_\_

Firma del Autor

**Liudmila Pupo Peña**

\_\_\_\_\_

Firma del Tutor

## **Datos de Contacto**

Ing. Liudmila Pupo Peña

Universidad de las Ciencias Informáticas, Habana, Cuba

Email: lpupo@uci.com

Graduado como Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2007. Profesor asistente con 14 años de experiencia en el trabajo con Gráficos por Computadoras y Realidad Virtual. Actualmente pertenece al Centro de Entornos Interactivos 3D VERTEX de la Facultad 4 de la UCI.

## AGRADECIMIENTOS

Agradezco a en primer lugar a mi familia por apoyarme en todas mis decisiones y brindarme su confianza. Agradezco a mi abuela por forjar mi trayectoria como estudiante. Gracias a mis amigos y la que considero es la familia más envidiada de la UCI, por su unidad, su creatividad y su talento, a la que todos les gusta llamar la familia #FP.

## **DEDICATORIA**

Esta tesis está dedicada a la sobrina más linda del mundo, del tío que más la quiere.

## RESUMEN

En el Centro de Entornos Interactivos 3D de la Universidad de la Ciencias Informáticas, en la rama de desarrollo dedicada a la Realidad Virtual se ha desarrollado el Paseo Virtual UCI. Esta aplicación permite recorrer las diferentes estructuras y lugares de la universidad brindando una información bastante completa. Se encuentra disponible una versión web, una versión de escritorio y una versión para dispositivos móviles con sistema operativo Android. Sin embargo, ninguna de estas versiones presenta sensación de inmersión y sus controles están restringidos solo al ratón y al teclado. La versión Android no utiliza las ventajas que brinda la tecnología móvil.

El uso de sensores como el giroscopio y la fácil portabilidad de los dispositivos pueden complementar y mejorar las funcionalidades de las aplicaciones que demandan interacción con el usuario, como lo hace un paseo virtual.

Para mejorar la inmersión del usuario, en la presente investigación se propone adaptar la versión Web ya existente del Paseo Virtual UCI a una versión Android, con vistas estereoscópicas y controles inteligentes, usando el visor *Cardboard*. La propuesta de solución emplea Unity como motor de videojuego, C# como lenguaje de programación y se rige por la metodología Programación Extrema.

Se obtuvo un módulo que garantiza el seguimiento de la posición del usuario, y genera vistas estereoscópicas compatibles con el visor *Cardboard*. Se realiza un cuestionario a 20 personas que concluye que la aplicación resultante presenta mayor inmersión y una experiencia de usuario más atractiva.

### PALABRAS CLAVE

*Cardboard*, Paseos Virtuales, Realidad Virtual, Visión Estereoscópica

# Tabla de Contenidos

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	4
1.1 Introducción .....	4
1.2 Paseos Virtuales .....	4
1.2.1 Antecedentes .....	4
1.2.2 Concepto de Paseo Virtual.....	5
1.2.3 Características de los Paseos Virtuales.....	6
A continuación, se muestran un grupo de características de los paseos virtuales:.....	6
1.3 Realidad Virtual.....	7
1.3.1 Concepto de Realidad Virtual .....	7
1.3.2 Características de la Realidad Virtual.....	7
1.3.3 La Realidad Virtual no Inmersiva .....	8
1.3.4 La Realidad Virtual Inmersiva.....	8
1.3.5 Especificaciones para el uso del sistema de Realidad Virtual.....	8
1.3.6 Aplicaciones de la Realidad Virtual.....	9
1.4 Visualización 3D.....	10
1.5 Modelado 3D.....	10
1.6 Estereoscopía .....	11
1.6.1 Definición de Estereoscopía .....	11
1.6.3 Técnicas de Visualización Estereoscópica .....	12
1.7 Motores Gráficos.....	13
1.7.1 Unity.....	13
1.7.2 Ventajas .....	14
1.7.3 Portabilidad .....	14
1.7.4 Asset Store e implementación de recursos.....	15
1.8 Visores estereoscópicos .....	15
1.8.1 HTC Vive.....	15
1.8.2 Oculus Rift.....	16



1.8.4 Gear VR .....	17
1.8.4 Cardboard .....	17
1.8.5 Comparación entre Visores .....	18
1.9 Metodología de desarrollo de software .....	18
1.9.1 Programación Extrema .....	19
1.10 Tecnologías a utilizar .....	20
1.10.1 Biblioteca .....	20
1.10.2 Lenguaje de Programación .....	21
1.10.3 Entorno de desarrollo integrado (IDE) .....	21
1.11 Conclusiones Parciales .....	21
<b>CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>23</b>
2.1 Introducción .....	23
2.2 Propuesta de Solución .....	23
2.3 Requisitos no funcionales .....	26
2.3.1 Captura de Requisitos no Funcionales .....	26
2.4 Fase de exploración .....	27
2.4.1 Historias de usuario .....	27
2.5 Fase de Planificación .....	30
2.5.1 Estimación de esfuerzo por Historias del usuario .....	30
2.5.2 Plan de iteraciones .....	31
2.5.3 Plan de duración de las iteraciones .....	32
2.5.4 Plan de entrega .....	32
2.6 Diseño de la solución propuesta .....	33
2.6.1 Arquitectura de Software .....	33
2.6.2 Patrones de diseño .....	34
2.6.3 Patrones GRASP .....	34
2.6.4 Patones GOF .....	34
2.6.5 Tarjetas CRC .....	34
2.7 Conclusiones Parciales .....	35
<b>CAPÍTULO 3: EVALUACIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>37</b>

3.1 Introducción .....	37
3.2 Estándar de Codificación .....	37
3.3 Fase de Producción .....	38
3.3.1 Pruebas.....	39
3.3.2 Entorno de Prueba .....	40
3.3.3 Pruebas de aceptación.....	40
3.4 Resultados Obtenidos .....	43
3.4.5 Encuesta de Aceptación.....	44
3.5 Conclusiones Parciales.....	46
<b>CONCLUSIONES</b> .....	47
<b>RECOMENDACIONES</b> .....	48
<b>ANEXOS</b> .....	49
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	52

## Índice de tablas

Tabla 1 Comparación entre tipos técnicas de estereoscopia .....	¡Error! Marcador no definido.
Tabla 2: Comparación entre visores estereoscópicos .....	18
Tabla 3: Estructura funcional del sistema.....	25
Tabla 4: Historia de usuario 1 .....	27
Tabla 5: Historia de usuario 2 .....	27
Tabla 6: Historia de usuario 3 .....	28
Tabla 7: Historia de usuario 4 .....	28
Tabla 8: Historia de usuario 5 .....	29
Tabla 9: Historia de usuario 6 .....	29
Tabla 10: Historia de usuario 7 .....	29
Tabla 11: Estimación de esfuerzo .....	30
Tabla 12: Plan de duración .....	32
Tabla 13: Plan de entrega.....	32
Tabla 14: Plan de entrega.....	33
Tabla 15: Tarjeta CRC 1 .....	34
Tabla 16: Tarjeta CRC 2 .....	35
Tabla 17: Tarjeta CRC 3 .....	35
Tabla 18: Tarjeta CRC 4 .....	35
Tabla 19: Tareas de Iteración 1 .....	39
Tabla 20: Tareas de ingeniería 1 .....	39
Tabla 21: Caso de prueba de aceptación 1.....	40
Tabla 22: Caso de prueba de aceptación 2.....	40
Tabla 23: Caso de prueba de aceptación 3.....	41
Tabla 24: Caso de prueba de aceptación 4.....	41
Tabla 25: Caso de prueba de aceptación 5.....	42
Tabla 26: Caso de prueba de aceptación 6.....	42
Tabla 27: Caso de prueba de aceptación 7.....	43
Tabla 28: Cuadro Lógico de ladov .....	45
Tabla 29: Tareas de ingeniería 2 .....	49
Tabla 30: Tareas ingeniería 3 .....	49

Tabla 31 : Tareas ingeniería 4 .....	49
Tabla 32 : Tareas ingeniería 5 .....	50
Tabla 33 : Tareas ingeniería 6 .....	50
Tabla 34 : Tareas ingeniería 7 .....	50

## Índice de figuras

Figura 1: Paseo Virtual "La playa" .....	6
Figura 2: Visión binocular .....	11
Figura 3: HTC Vive .....	16
Figura 4: Oculus Rift.....	16
Figura 5: Gear VR.....	17
Figura 6: Cardboard .....	18
Figura 7: Propuesta de solución.....	24
Figura 8: Estructura funcional del sistema.....	25
Figura 10: Diseño arquitectónico del sistema .....	33
Figura 10: Fragmento de código .....	38
Figura 11: Vista estereoscópica de la escena .....	44
Figura 12: Vista de usuario con el visor Cardboard .....	44
Figura 13: Encuesta a usuarios sobre producto final.....	45
Figura 14: Eje de satisfacción (Técnica ladov) .....	46

# INTRODUCCIÓN

Con el surgimiento del Internet y el constante aumento del interés de las personas por conocer sobre nuevos lugares, que no siempre son tan fáciles de acceder, surgieron muchas formas que, desde una computadora, permiten sentir, de manera virtual, la sensación de visitar un lugar recibiendo su información visual lo más completa posible. Una de las aplicaciones más populares que lo hacen posible, son las denominadas Paseos Virtuales.

El Centro de Entornos Interactivos 3D (VERTEX) de la Universidad de la Ciencias Informáticas (UCI), tiene entre sus principales líneas de investigación el desarrollo de paseos virtuales. Uno de sus principales productos en este tema es el Paseo Virtual de la UCI.

El Paseo Virtual de la UCI es una aplicación que permite mostrar el contenido visual de los espacios de la UCI como son los docentes estudiantiles, las esculturas, las áreas deportivas y las áreas de servicio. Es una herramienta tecnológica mediante la cual el observador puede trasladarse de forma interactiva de un espacio a otro, dándole la sensación de recorrer la infraestructura y permitiéndole conocer detalles de su distribución y características.

En este Paseo Virtual no existe una percepción de inmersión, porque a pesar de tener un diseño en 3D, el observador recibe el contenido visual por medio de una pantalla, sin ninguna técnica de visualización estereoscópica. También, para la navegación, por la escena se usan el mouse y el teclado. La versión web, a pesar de ser multiplataforma, limita los gráficos del diseño, y la versión de escritorio solo está destinada a la computadora.

La aplicación cuenta con una versión para teléfonos móviles con sistema operativo Android, aprovechando que cada vez son más las personas que apuestan por la tecnología móvil, dejando en muchos casos las computadoras de escritorio como su segunda opción. El desarrollo tecnológico de estos dispositivos, tanto en el hardware, como en el software, llevan las aplicaciones Android a otro nivel.

Sin embargo, en la versión disponible hasta el momento no se aprovechan todas las ventajas que brindan los dispositivos móviles. Una de las formas novedosas de vincular los Paseos Virtuales con los móviles, aprovechar al máximo sus ventajas y crear una sensación inmersión, puede hacerse con el uso de uno de los dispositivos de menos costo del mercado de la Realidad Virtual (RV), el *Google Cardboard* (1).

La Realidad Virtual (RV) exige un hardware muy costoso, por lo que Google propone un dispositivo casero y económico, que aprovecha el potencial de los teléfonos Android, basados en cartón y dos lentes de cristal.

A partir de lo anteriormente mencionado se llega al siguiente **problema de investigación**:

- ¿Cómo lograr una mayor inmersión en el Paseo Virtual UCI?

El **campo de acción** está enmarcado por el empleo de *CardBoard* en paseos virtuales.

El **objeto de estudio** se concentra en los métodos de visualización estereoscópica.

El **objetivo** es desarrollar el Paseo Virtual UCI usando la tecnología Cardboard para dispositivos móviles con sistema operativo Android.

Para una mejor comprensión, el documento está dividido en capítulos que estructuran el contenido de la siguiente forma:

**Capítulo 1:** “Fundamentación Teórica”. Constituye la base teórica de la investigación realizada. Se describen los principales conceptos relacionados con las formas de lograr una interactividad entre el usuario y una Aplicación RV. Además, se describen las herramientas y metodologías utilizadas para el desarrollo del software.

**Capítulo 2:** “Solución Propuesta”. Se describe la propuesta de solución teniendo en cuenta el marco teórico, la metodología y las herramientas a utilizar para el desarrollo de software. Además, se muestran los documentos generados por la metodología empleada.

**Capítulo 3:** “Implementación y validación de la solución propuesta”. Se exponen los detalles correspondientes a la implementación del sistema. También se describen las pruebas realizadas al sistema y los principales resultados obtenidos.

Los métodos científicos de investigación que se tendrán en cuenta para el desarrollo de esta investigación son los siguientes:

### **Métodos Teóricos**

- **Histórico-Lógico:** Seleccionado para llevar a cabo un análisis valorativo de la bibliografía existente, así como para conocer las tendencias actuales en el campo de la RV y el desarrollo de paseos virtuales para dispositivos móviles. Se analiza el estado del arte existente respecto a la problemática planteada y se selecciona la bibliografía, así como las herramientas y/o bibliotecas más utilizadas para este fin.
- **Analítico-Sintético:** Se utiliza este método científico de investigación para poder descomponer el problema en partes más concretas y llegar a conclusiones más específicas, y luego lograr la correcta integración y comunicación entre sí.

## Métodos Empíricos

- **Entrevista:** La entrevista a especialistas con experiencia en el empleo de las herramientas como motores de videojuegos y el trabajo con dispositivos móviles con sistema operativo Android.

**Consulta de fuentes de Información:** Permitted determinar el estado del arte del objeto de investigación, consultando una bibliografía completa y actualizada.

Para dar cumplimiento al objetivo planteado se trazan las siguientes **tareas investigativas:**

- Definir las características y los antecedentes de los Paseos Virtuales para entender su importancia y necesidad.
- Definir las técnicas de visualización 3D para escoger la más adecuada para dispositivos móviles.
- Definir las técnicas y características de estereoscopía.
- Definir las características y ventajas que brinda Unity para el desarrollo de aplicaciones de RV para decidir si es posible y recomendable utilizar el motor gráfico.
- Definir las características del dispositivo *Cardboard* para entender su uso y funcionamiento.
- Plasmar todos los recursos ingenieriles necesarios y la metodología de investigación a seguir.
- Definir una propuesta de solución para la implementación de la aplicación.
- Validar el resultado de la aplicación obtenida para garantizar un producto libre de errores e inconformidades.



## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

Es indiscutible el fuerte impacto que ha tenido en la actualidad la RV, y aunque su desarrollo es innovador, los efectos en la adaptación de las personas a los cambios que la ciencia informática impone son considerables. Las sensaciones que pueden lograrse mediante la RV llegan a extremos muy cercanos a la realidad, mediante los dispositivos electrónicos capaces de procesar esta tecnología, sitúan al usuario en un ambiente casi real y permite interactuar con lo que le rodea (2).

Los paseos Virtuales son una de las formas más claras de aplicar la RV, permiten que al usuario le llegue la información lo más completa posible, de un lugar o monumento de su interés. Para su implementación es necesario el conocimiento de un grupo de elementos vinculados con el éxito de su desarrollo, como el tipo de RV y las técnicas para conseguirla.

En el presente capítulo, se presenta un estudio del estado del arte referente al desarrollo de paseos virtuales, para ello se exponen una serie de conceptos y temas relacionados. Por último, se hace referencia a diferentes enfoques utilizados por los desarrolladores de este tipo de productos a nivel mundial, para dar solución a este problema.

### **1.2 Paseos Virtuales**

Un Paseo Virtual es uno de los métodos más utilizados para lograr que al usuario le llegue una información visual lo más completa posible respecto a un lugar. Resulta de mucha utilidad para el usuario, ya que todo puede hacerse mediante un proceso virtual y perfectamente realizable desde su propia habitación (3).

#### **1.2.1 Antecedentes**

No se tiene conocimiento certero que indique una fecha específica del momento en que apareció la primera aplicación basada en la presentación de contenido visual y con entornos físicos de un determinado lugar, sin embargo, se sabe que los pioneros en la creación de este tipo de aplicaciones fueron las empresas de aeronáutica comercial o comúnmente conocidas como aerolíneas, de Estados Unidos y del continente Europeo, quienes entre los años 1996 y 1997 empezaron a apoyarse en la presentación de imágenes, en sus respectivos sitios web, de los diferentes destinos a donde ofrecían sus servicios de aviación comercial a los viajeros (4).

Con el pasar del tiempo y a la vez del desarrollo de nuevas herramientas tecnológicas, este tipo de instrumentos fue evolucionando, de manera que más empresas y sectores económicos fueron valiéndose

de los Paseos Virtuales como una herramienta de apoyo tecnológico para dar a conocer sus servicios a las personas. Entre las empresas y sectores económicos que se unieron al uso de este tipo de herramienta están: hoteles, restaurantes, operadoras de viajes, museos, parques de diversión, instituciones educativas, empresas constructoras y aerolíneas.

### **1.2.2 Concepto de Paseo Virtual**

Un Paseo Virtual es una efectiva forma de mostrar el contenido visual sobre un espacio o una infraestructura, es una herramienta tecnológica mediante la cual el observador se traslada interactivamente de un espacio a otro, dándole la sensación de recorrer la infraestructura y permitiéndole conocer a detalle su distribución y características (5).

Los usuarios de un Paseo Virtual pueden percibir con mayor intensidad con respecto a la información que se obtiene de una imagen o video, las dimensiones y emplazamiento de la infraestructura que se muestra, además se puede realizar un recorrido por todas o parte de las instancias de un determinado espacio o ubicación de un lugar; por ejemplo: una casa, un edificio, un parque, una universidad, un restaurante, un hotel, un espacio al aire libre o un museo.

Todo esto desde un medio electrónico que permite acceder al Paseo Virtual y mostrar la infraestructura de un determinado lugar que se desee conocer sin necesidad de estar físicamente presente, necesitando solamente un ordenador o con otro tipo de dispositivo que soporte una aplicación de ese tipo, pudiendo así hacer el recorrido por el Paseo Virtual desde la comodidad de su casa a cualquier lugar o parte del mundo como se muestra en la Figura 1.



**Figura 1: Paseo Virtual "La playa"**

### **1.2.3 Características de los Paseos Virtuales**

A continuación, se muestran un grupo de características de los paseos virtuales:

- Herramienta tecnológica innovadora.
- Presentación de imágenes cilíndricas, esféricas o mapa de objetos en 3D.
- Visión total del espacio.
- Imágenes sin distorsión.
- Presenta una percepción real del espacio.
- Incrementa el nivel de información de los clientes o turistas potenciales.
- Permite visualizar cualquier tipo de espacio físico o lugar.
- Aumenta la satisfacción en el proceso de apreciación de una infraestructura.
- Capacidad de incitar y generar el deseo en el espectador de estar físicamente en el lugar que se está visualizando.
- Fácil acceso mediante Internet que requiere poco conocimiento de informática.
- Interfaz interactiva que incluye imágenes y elementos multimedia de gran atractivo visual.
- Empleo de aplicaciones y herramientas de fácil uso para el espectador.
- Fácil actualización y modificación del contenido (5).

### **1.3 Realidad Virtual**

Un mundo virtual permite realizar actividades especiales, el usuario puede incursionar creativamente, hasta donde el límite de su imaginación se lo permita. Se vuelve realmente atractivo, mientras más real parezca el mundo virtual, mayor aceptación.

#### **1.3.1 Concepto de Realidad Virtual**

Es todo aquello que tiene que ver con imágenes en tres dimensiones generadas por ordenador y con la interacción de los usuarios con este ambiente gráfico. Ello supone la existencia de un complejo sistema electrónico para proyectar espacios visuales en 3D y para enviar y recibir señales con información sobre la actuación del usuario, quien, con un sistema de este tipo, puede sentir que se encuentra inmerso en un mundo virtual (6).

Un mundo virtual es un modelo matemático que describe un espacio tridimensional, que contiene objetos que pueden representar cualquier cosa. Desde una simple entidad geométrica, por ejemplo, un cubo o una esfera, hasta una forma compleja, como puede ser un desarrollo arquitectónico, un nuevo estado físico de la materia o el modelo de una estructura genética. Se trata, en definitiva, de un paso más allá de los que sería la simulación por computadora, tratándose realmente de la simulación interactiva, dinámica y en tiempo real de un sistema.

#### **1.3.2 Características de la Realidad Virtual**

A continuación, se muestran una serie de reglas y características para un correcto desarrollo y aceptación del usuario que sigue la RV.

- Se expresa en lenguaje gráfico tridimensional.
- Su comportamiento es dinámico y opera en tiempo real.
- Su operación está basada en la incorporación del usuario en el interior del medio computarizado.
- Requiere que, en principio haya una suspensión de la incredulidad como recurso para lograr la integración del usuario al mundo virtual al que ingresa.
- Posee la capacidad de reaccionar ante el usuario, ofreciéndole, en su modalidad más avanzada, una experiencia inmersiva, interactiva y multisensorial.

### 1.3.3 La Realidad Virtual no Inmersiva

“También conocida como RV de escritorio, es donde el usuario puede simular una RV a través de la ventana de un monitor y los accesorios utilizados como medio de interacción son el teclado, ratón, micrófono o palanca de juegos, este tipo de RV es ideal por su bajo costo para trabajos científicos y también es un modo de entretenimiento” (7).

Este tipo de RV es el más común, no ofrece una sensación de inmersión y no necesita de ningún dispositivo de visualización. Actualmente el Paseo Virtual UCI presenta una RV no inmersiva (7).

### 1.3.4 La Realidad Virtual Inmersiva

“En la Realidad Virtual inmersiva el usuario se siente dentro del mundo virtual que está explorando. En este tipo de RV se utilizan diferentes dispositivos denominados accesorios los cuales pueden ser guantes o trajes especiales, un casco visor o *Helmet mounted display* (HMD) que permite al usuario visualizar a través de ellos y a la vez aislarse en cierto modo del mundo exterior, la idea de este tipo es la de hacer que el usuario vea algo irreal como si no lo fuera, esto hace posible la aplicación de entretenimiento y capacitación”.

La inmersión en este campo le brinda al usuario una abstracción perfecta para el tipo de aplicación que es un paseo virtual, haciendo que la información visual además de ser bastante completa sea agradable y entretenida (7).

### 1.3.5 Especificaciones para el uso del sistema de Realidad Virtual

Las especificaciones de cada uno de los pasos evolutivos en el acercamiento al uso de la RV son:

- **Teoría:**

Los conocimientos requeridos para el manejo de herramientas de RV pueden y deben darse como primer paso en las actividades, siendo indispensable establecer una sólida base inicial de conocimientos aún antes de proceder a adquirir el equipamiento requerido.

- **Simulación 3D gráfica:**

Se refiere al uso interactivo de programas, lo cual ofrece, en la actualidad, un amplio rango de opciones según la capacidad adquisitiva y el nivel de experimentación en que se busca, desde programas gratuitos (*freeware*) que se ofrecen en Internet hasta programas comerciales como el *VirtusWalkrough Pro*. Estas opciones no incluyen la inmersión.

- **Simulación estéreo:**

Implica la experimentación con aspectos de inmersión. Puede evolucionar desde el uso de lentes sencillos tipo Sega hasta cascos del tipo HMD. También puede complementar, en su nivel superior de costos, la inmersión visual con la de sonido estereofónico por medio de sofisticados recursos.

- **Simulación háptica:**

Este aspecto concierne a la información accesible a través del tacto, ya sea tocando o manipulando objetos con percepción de consistencia, textura y, en los casos más avanzados, de resistencia y peso. Abarca desde el uso de guantes de bajo costo tipo Nintendo hasta el uso de recursos sofisticados y costosos del tipo *Dataglove* (VPL), o de los denominados trajes de datos.

- **Realidad virtual integral:**

Se refiere a la integración, dentro de un mismo ambiente, de los diferentes tipos de simulación anteriormente mencionados, con el objetivo de generar al máximo la ilusión de realidad. Representa una situación ideal a la cual aspirar como experimentador.

### **1.3.6 Aplicaciones de la Realidad Virtual**

En la actualidad, la RV se plasma en una multiplicidad de sistemas. El más conocido es el que ha desarrollado la empresa norteamericana VPL, con la que la NASA trabaja en estrecha colaboración en el desarrollo de sus propias aplicaciones.

Se desarrolló una arquitectura básica para la creación de una variedad casi ilimitada de laboratorios virtuales. En ellos, los científicos de disciplinas muy diversas son capaces de penetrar en horizontes antes inalcanzables gracias a la posibilidad de estar ahí: dentro de una molécula, en medio de una violenta tormenta o en una galaxia distante (8).

Profesionales de otros campos utilizan los laboratorios virtuales para una gran variedad de funciones, a continuación, se muestran diferentes aplicaciones dentro del campo profesional con excelentes resultados: Los cirujanos pueden realizar operaciones simuladas para ensayar las técnicas más complicadas, antes de una operación real.

En ingeniería se desarrollan aplicaciones para la aereo-industria, industria automovilística en modelos electrónicos de vehículos para probar confort.

En el ámbito científico, investigadores de la Universidad de Carolina del Sur estudian moléculas complejas, desplazando grupos de átomos mediante un instrumento, una simbiosis entre los punteros del tipo del ratón y el *Dataglove*.

En educación y adiestramiento se da la exploración de lugares y cosas inaccesibles por otros medios. Creación de lugares y cosas con diferentes cualidades respecto a los que existen en el mundo real (9). Interacción con o tras personas, ubicadas en áreas remotas, de intereses afines. Colaboración en la realización de proyectos con estudiantes alrededor del mundo.

#### **1.4 Visualización 3D**

El poder apreciar las tres dimensiones de los objetos, se debe a la capacidad del cerebro de interpretar la información administrada por los ojos desde ángulos diferentes dando paso a la percepción de profundidad. Haciendo uso de dos imágenes que enfocan un mismo objeto desde dos ángulos distintos, técnica llamada estereoscopia. Desde 1840 se logra emular la visión con la ayuda de lentes especiales o métodos de visualización, siendo únicamente posible la percepción tridimensional de escenas estáticas.

#### **1.5 Modelado 3D**

Un modelo en 3D es un mundo conceptual en tres dimensiones que puede observarse de dos formas distintas. Desde un punto de vista técnico, es un grupo de fórmulas matemáticas que describen un mundo en tres dimensiones. Desde un punto de vista visual, valga la redundancia, un modelo en 3D es una representación esquemática visible a través de un conjunto de objetos, elementos y propiedades que, una vez procesados se convertirán en una imagen en 3D o una animación 3D.

Por lo general, el modelo visual suele ser el modelo 3D que los diseñadores manejan, dejando las fórmulas a procesos computacionales. Esto es así, porque lo que el modelo en 3D visual representa se acerca más a la imagen en 3D final que se mostrará al renderizarse (9).

Existen aplicaciones de modelado en 3D, que permiten una fácil creación y modificación de objetos en tres dimensiones. Estas herramientas suelen tener objetos básicos poligonales como esferas, triángulos, cuadrados para ir armando el modelo. Además, suelen contar con herramientas para la generación de efectos de iluminación, texturizado, animación y transparencias. Algunas aplicaciones de modelado son *3D Studio Max, Alias, Blender, Cheetah 3D, Cinema 4D, GenerativeComponents, Houdini, LightWave, Maya, MilkShape 3D, modo Rhinoceros 3D, Softimage|XSI, trueSpace* y *ZBrush*.

El modelo en 3D describe un conjunto de características que, en conjunto, resultarán en una imagen en 3D. Este conjunto de características suele estar formado por objetos poligonales, tonalidades, texturas, sombras, reflejos, transparencias, translucidez, refracciones, iluminación (directa, indirecta y global), profundidad de campo, desenfoques por movimiento, ambiente, punto de vista.

## 1.6 Estereoscopía

El objetivo básico de la estereoscopía, que se remonta a los inicios de la fotografía, es intentar obtener una imagen en relieve, es decir que de sensación de tres dimensiones con la que sea posible percibir la profundidad. A lo largo de la historia se han ido desarrollando diversas técnicas, para conseguir esta sensación de 3D

### 1.6.1 Definición de Estereoscopía

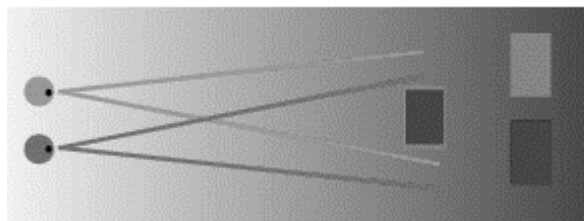
“La palabra estéreo etimológicamente proviene del griego y significa relativo al espacio, la estereoscopía, o proyección de imagen tridimensional, es la visión que posee el hombre, cuyo cerebro recrea una integración a escala de las imágenes recogidas por la retina del ojo izquierdo y derecho simultáneamente interpretando así lo que se ve” (10).

Para que el cerebro pueda percibir una imagen en tercera dimensión, requiere de datos sobre la distancia de los objetos, dicha información se obtiene gracias a que se tienen dos ojos con una debida separación, así cada uno percibe los objetos desde un ángulo distinto, dando como resultado una triangulación de la cual el cerebro obtiene la distancia al objeto. La reconstrucción de la profundidad a partir de dos imágenes se conoce como estereopsis.

La información tridimensional de la profundidad se puede reconstruir a partir de dos imágenes o usando una computadora y con un software que permita crear la información significativa de la profundidad.

Si se toman o se crean dos imágenes con un ángulo ligeramente distinto y se le muestran a cada ojo por separado, el cerebro podrá reconstruir la distancia y recrear la profundidad.

De aquí se extrae la conclusión de que las variaciones horizontales que hacen que las imágenes tengan un ángulo ligeramente diferente pueden ser interpretadas por nuestro cerebro como una realidad con volumen (10).



**Figura 2: Visión binocular**



El ojo percibe los objetos en diferentes ángulos, creando la ilusión de profundidad de los objetos. Las variaciones verticales son indiferentes en lo que respecta a creación de sensación de volumen (a no ser que esta diferencia sea demasiado grande).

Esta tercera dimensión es capaz de reconstruirse en el cerebro gracias a una serie de complejos procesos fisiológicos y psicológicos relacionados con la visión tanto monocular como la binocular:

**Visión monocular:** Cuando se mira solo con un ojo y se crea una imagen plana en 2D, pero con información intuitiva de profundidad y de distancia.

**Visión binocular:** Cuando se mira con ambos ojos, esta es la que aporta una mayor información espacial y por tanto de la tercera dimensión permitiendo la creación de la sensación de volumen.

### 1.6.3 Técnicas de Visualización Estereoscópica

Para la visualización o proyección estereoscópica existen dos tipos de técnicas, las de tipo Pasivo y las de tipo Activa. Las técnicas de visualización Pasiva más conocidas son Anaglifos y Polarizadas y las técnicas de visualización activa más conocidas son las *ShutterGlasses* y la Lenticular, a continuación, una tabla con sus características (10).

**Tabla 1: Comparación entre tipos técnicas de estereoscopia**

Tipo	Activa	Pasiva	Características	Ventajas/Desventajas	costo
<b>Anaglifo</b>		X	Filtros de colores (gafas) Trabaja con todas las pantallas	suprime el color	Bajo
<b>Polarizado</b>		X	Uso de 2 monitores o proyectores Esquema complejo de proyección	no suprime el color	Moderado
<b>ShutterGlasses</b>	X		Doble imagen Modo complejo de proyección No hace el efecto <i>ghost</i> (fantasma)	baja resolución mejor proyección	Alto

<b>Lenticular</b>	X		No requiere gafas -8 cámaras de punto de vista -programación compleja	baja resolución mejor proyección	Alto
<b>Visión Paralela</b>		X	Requiere lentes para que cada ojo vea la imagen que le corresponde.	media resolución mejor proyección	Bajo

## 1.7 Motores Gráficos

En la industria de los videojuegos se hace uso de los Motores de videojuegos como herramientas tecnológicas para el desarrollo de un proyecto, cada uno de estos ofrece un conjunto de componentes haciendo que la complejidad de los procesos de desarrollo sea menor. Estos componentes poseen diferentes implementaciones en cada uno de los motores, haciendo que estas diferencias creen una mirada objetiva sobre el uso de estos a partir de las necesidades que se presenten en los proyectos (11).

### 1.7.1 Unity

Unity es un motor de videojuegos multiplataforma desarrollado por *Unity Technologies*, que permite desarrollar para videoconsolas, PC, Mac, GNU/Linux, todo tipo de dispositivos móviles, la web, televisores y dispositivos de RV.

En un principio la propia herramienta fue anunciada únicamente para el sistema operativo OS X, sin embargo, desde su lanzamiento en 2005 ha aumentado su soporte a más de 25 plataformas. Por este motivo, además de otros que se mencionarán a continuación, Unity es la herramienta a utilizar en el proyecto.

Al ser un motor de videojuegos, Unity proporciona un conjunto de componentes y herramientas comunes listas para ser utilizadas, como por ejemplo *shaders*, simulación de físicas o efectos de iluminación, entre otros. Esto puede facilitar considerablemente la implementación respecto a un videojuego o aplicación desarrollada "desde cero", ya que no es necesario invertir tiempo en la preparación de estos componentes, que pueden ser necesarios o no según el tipo de aplicación que se esté desarrollando.

Unity permite desarrollar tanto en 3D como en 2D, además de proporcionar soporte para juegos y aplicaciones de RV y juego online multijugador (12).

### 1.7.2 Ventajas

Hay varias características que diferencian a Unity de otras alternativas, entre las que destacan especialmente un editor visual que facilita hacer cambios y probarlos de una forma muy productiva, y una gran portabilidad que permite generar ejecutables para casi cualquier plataforma, entre otras (11).

### 1.7.3 Portabilidad

Otra de las razones de la rápida adopción de Unity como herramienta de desarrollo es su capacidad para generar ejecutables para múltiples plataformas, incluyendo las principales videoconsolas, ordenadores personales y dispositivos móviles. La lista ha ido aumentando con el tiempo y con la aparición de nuevo hardware. A continuación, se muestra la lista de todas las plataformas soportadas oficialmente en la actualidad (11).

Ordenadores personales y Web:

- *Windows*
- Mac OS.
- Linux/Steam OS
- Web GL
- Facebook Gameroom

#### Dispositivos móviles

- iOS
- Android
- Windows Phone
- Tizen
- Fire Os.
- Windows Apps

#### Realidad Virtual

- Oculus Rift
- Google *Cardboard*
- Steam VR
- Playstation VR
- Gear VR
- Microsoft Hololens

- Daydream.

#### **1.7.4 Asset Store e implementación de recursos**

La Asset Store es un repositorio de contenidos creados por la comunidad que pueden importarse directamente en un proyecto de Unity, como modelos, imágenes o sonidos. Algunos de estos recursos son gratuitos y otros son de pago, y existen varias licencias de uso distintas según el contenido. La utilización de estos recursos puede facilitar el desarrollo considerablemente (12).

### **1.8 Visores estereoscópicos**

En esta sección se expone el estudio de múltiples opciones hardware de RV existentes en la actualidad. Se evalúa el abanico de alternativa y se escoge el resultado más óptimo para el desarrollo de la propuesta.

#### **1.8.1 HTC Vive**

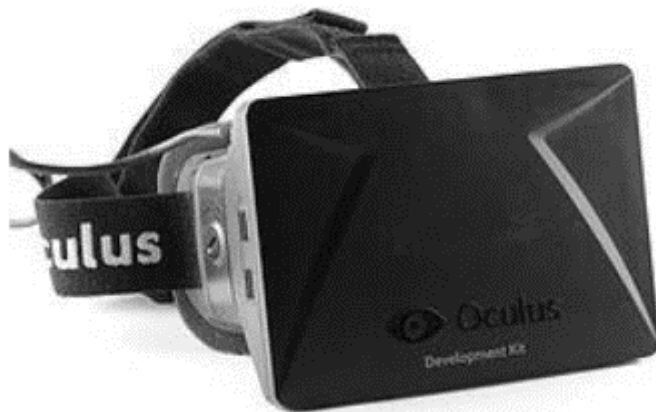
Es desarrollado por HTC y Corporación Valve y fue lanzado en abril de 2016. A diferencia de la versión actual del Oculus Rift, el HTC Vive está diseñado para convertir una habitación en un espacio 3D. Dos unidades de referencia fijas rastrean la cabeza del usuario y los controladores de mano, mientras que el usuario puede caminar y manipular objetos virtuales. El HTC Vive utiliza un diodo de visualización y emisores de luz orgánicos (OLED) y proporciona una resolución combinada de 2160x1200 (1080x1200 por ojo) con una frecuencia de actualización de 90 Hz y un campo de visión (FOV) de aproximadamente 110 grados. La pantalla montada en la cabeza pesa aproximadamente 555 gramos y tiene conexiones HDMI 1.4, DisplayPort 1.2 y USB 2.0. Además, el HTC Vive tiene un conector de audio de 3,5 mm para auriculares y un micrófono incorporado. Finalmente, el HTC Vive tiene una cámara frontal para combinar elementos del mundo real en el mundo virtual (13).



**Figura 3: HTC Vive**

### **1.8.2 Oculus Rift**

El dispositivo de Oculus ha ido recibiendo gran cantidad de mejoras desde que apareció el primer prototipo en 2012, ofreciendo actualmente unas características muy superiores a las de su primera versión, ofreciendo, en su versión *Crystal Cove* (Enero de 2014), la opción de 1080p, junto al uso del primer display con tecnología OLED del mercado y un nuevo sistema de detección del movimiento mediante una cámara externa que detecta la posición de unos puntos infrarrojos situados en el dispositivo, lo que permite al sistema detectar acciones del usuario tales como agacharse (14).



**Figura 4: Oculus Rift**

#### 1.8.4 Gear VR

El *Gear VR* es un equipo de realidad virtual para juegos electrónicos desarrollado por Samsung, que es compatible con algunos dispositivos móviles, como Samsung Galaxy S7 Edge, Galaxy S7, Galaxy Note 5 y Galaxy S6. Usa los dispositivos móviles para llevar las aplicaciones 3D al siguiente nivel. Proyecta las imágenes en una pantalla LCD con la ayuda del acelerómetro y el giroscopio reconoce los movimientos de la cabeza del usuario proporcionados por el dispositivo móvil. Los parámetros de sensibilidad de tacto, comodidad de la vista y volumen pueden ajustarse sin retirarse de la cabeza del usuario. Tiene sensores de proximidad para ahorrar la batería si no está siendo usado y no necesita ser cargado al consumir energía del teléfono inteligente (15).



**Figura 5: Gear VR**

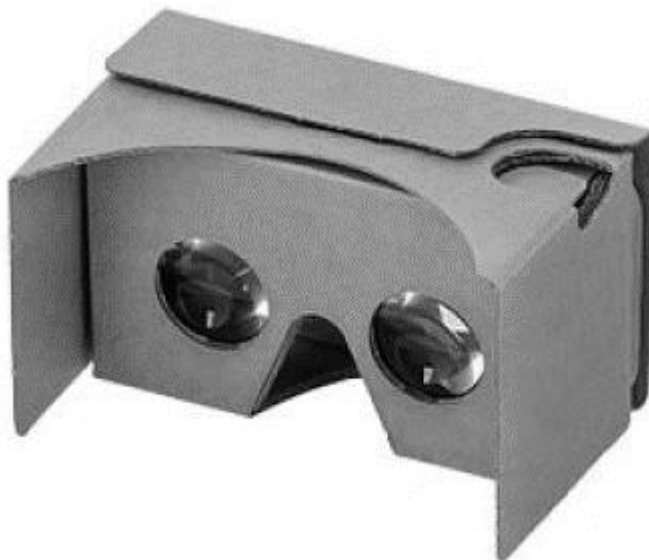
#### 1.8.4 Cardboard

“Google *Cardboard* es una plataforma desarrollada por Google y destinada a emplear como visor cualquier teléfono compatible junto con un soporte de cartón que incluye dos lentes y un gatillo magnético (figura a). Su principal ventaja es que es una forma accesible y económica de obtener un dispositivo de RV, ya que cualquier persona puede realizar el montaje del visor a partir de los materiales necesarios.

Tras montar el soporte de cartón con las lentes, el teléfono se introduce en la parte posterior. Las aplicaciones desarrolladas para esta plataforma dividen la pantalla en dos y aplican un efecto de distorsión que, junto con el efecto de las lentes, produce una imagen estereoscópica con un campo de visión bastante amplio”.

Existen muchos modelos de visor disponibles, y cada uno determina los requisitos necesarios del teléfono que se vaya a utilizar, si bien en la mayoría de los casos es necesario que el terminal tenga giroscopio y

sensor magnético para poder detectar las pulsaciones del gatillo. En cuanto al sistema operativo del teléfono, se requiere Android 4.4 o bien iOS 8.0 (o versiones posteriores) (16).



**Figura 6: Cardboard**

### 1.8.5 Comparación entre Visores

En la siguiente tabla se muestra una comparación entre los visores estereoscópicos que puede formar parte de la propuesta de solución. Se toman en cuenta, el precio, la tecnología a la que está destinada, y la portabilidad; dando como resultado que el dispositivo más adecuado para la solución sea el cardboard.

**Tabla 2: Comparación entre visores estereoscópicos**

<i>Nombre</i>	<i>Destino</i>	<i>Precio</i>	<i>Portabilidad</i>
<b><i>HTC Vive</i></b>	Computadoras	Alto	Baja
<b><i>Oculus Rift</i></b>	Computadoras	Alto	Baja
<b><i>Gear VR</i></b>	Teléfonos inteligentes	Medio	Alta
<b><i>Cardboard</i></b>	Teléfonos inteligentes	Bajo	Alta

### 1.9 Metodología de desarrollo de software

Una metodología es el conjunto de técnicas y procedimientos que permiten conocer los elementos necesarios para definir un proyecto de software, es la base para la edificación de un producto de este tipo. Esencialmente, controlar de manera transparente todo el proceso de desarrollo. Si se quiere que un proyecto

sea escalable y flexible a los cambios, se recomienda tomar en cuenta una metodología de desarrollo de software. Las metodologías, dadas sus características, se enmarcan en dos grandes grupos: las llamadas "metodologías pesadas" y las "metodologías ágiles". La diferencia más notable entre estos dos grupos es que mientras los métodos pesados intentan obtener los resultados apoyándose principalmente en la documentación ordenada, los métodos ágiles tienen como base de sus resultados la comunicación e interacción directa con todos los usuarios involucrados en el proceso.

### 1.9.1 Programación Extrema

Durante todo el proceso de desarrollo de software los objetivos fundamentales son que el producto final tenga la calidad requerida y que cumpla con las exigencias del cliente. Para lograr que estos objetivos se realicen es necesario utilizar una metodología de desarrollo de software, para escogerla hay que basarse en las características específicas que tiene cada producto.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Se basa en la comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y agilidad para enfrentar los cambios. *XP* es especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Se plantea que el ciclo de vida ideal de *XP* consiste de seis fases: Exploración, Planificación de la Entrega (*Release*), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto (17).

El ciclo de desarrollo se define en los siguientes pasos:

- El cliente define el valor de negocio a implementar.
- El programador estima el esfuerzo necesario para su implementación.
- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- El programador construye ese valor de negocio.
- Vuelve al paso 1.

Las características esenciales de *XP* son: historias de usuario (HU), roles, procesos y prácticas. Donde las HU constituyen tarjetas para especificar las necesidades del software, son comprensibles y delimitadas para que los programadores puedan implementarlas en poco tiempo. Plantea roles como: programador, cliente, encargado de pruebas, encargado de seguimiento, entrenador, consultor y gestor. El proceso no es más que el ciclo de vida y las prácticas consisten en la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo necesario para que el diseño evolutivo funcione.



La metodología *XP* se caracteriza por realizar entregas pequeñas y frecuentes, producto de dichas pautas, se puede llevar un control de las tareas y se obtienen resultados en muy poco tiempo. Además, no exige una amplia documentación del trabajo realizado.

Debido a que el autor dispone de un corto tiempo para la entrega final de la solución, es el único miembro del equipo de desarrollo de la presente investigación; en la que los requisitos corren el riesgo de ser inestables y consecuentemente se realizarán pequeñas entregas para evaluar el avance de la solución, se decide utilizar la metodología *XP* para guiar el proceso de desarrollo de la solución.

## **1.10 Tecnologías a utilizar**

Un elemento importante a la hora de desarrollar un software, lo constituye las herramientas a utilizar, las cuales pueden contribuir de una forma importante a mejorar la efectividad del mismo, así como el tiempo de desarrollo.

### **1.10.1 Biblioteca**

La integración nativa de Unity con Google VR facilita la creación de aplicaciones de Android para *Daydream* y *Cardboard*. El SDK de Google VR para Unity proporciona características adicionales como audio especializado, compatibilidad con el controlador *Daydream*, utilidades y características (18).

El soporte nativo de Unity para Google RV hace fácil que:

- Comience desde cero un nuevo proyecto RV Unity.
- Adapte una aplicación existente de Unity a RV.
- Crea una aplicación que pueda cambiar fácilmente de modo RV.

La integración con Google VR proporciona:

- Seguimiento del usuario.
- Representación estéreo lado a lado.
- Detectando la interacción del usuario con el sistema (a través del disparador o controlador).
- Corrección de distorsión para lentes de un visor RV.
- Un marcador de alineación para ayudar a centrar la pantalla debajo de los lentes cuando se inserta un teléfono en un visor.
- Corrección derivada automática del giroscopio.

El SDK de Google VR para Unity ofrece las siguientes características adicionales:

- Soporte de controlador *Daydream*.

- Representación de audio especializado.
- Un simple retículo prefabricado y scripts asociados a la interacción del usuario basada en la mirada.
- Emulación de RV en el modo de reproducción del editor de Unity, usando el ratón y las teclas alt / control para desplazar o inclinar la cámara.
- Una escena de “Demostración de auriculares” que muestra un juego simple de cartón y una escena de “demostración de controlador” integrada con el controlador *Daydream*.
- Una pantalla *frame per second* (FPS) prefabricada que muestra el rendimiento de representación de su aplicación.
- La herramienta Vista previa instantánea, una aplicación prefabricada que le permite obtener una vista previa de una escena en el dispositivo sin tener que compilarla e implementarla.

### 1.10.2 Lenguaje de Programación

La utilización de la biblioteca *Google VR*, la visualización estereoscópica, motiva a utilizar como lenguaje base *C#*.

En la actualidad, *C#* es un lenguaje elegante, con seguridad de tipos y orientado a objetos, que permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas que se ejecutan en .NET Framework .NET (19). Puede usar *C#* para crear aplicaciones cliente de Windows, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y muchos otros programas informáticos. Visual *C#* proporciona un editor de código avanzado, prácticos diseñadores de interfaz de usuario, un depurador integrado y muchas otras herramientas que facilitan el desarrollo de aplicaciones basadas en el lenguaje *C#* y .NET Framework.

### 1.10.3 Entorno de desarrollo integrado (IDE)

En la investigación se utilizará Visual Studio 2012 como IDE de desarrollo. Este IDE facilita el trabajo con el lenguaje de programación *C#* y tiene una integración con Unity que le permite usar todas las bibliotecas relacionadas con el proyecto que se esté desarrollando.

### 1.11 Conclusiones Parciales

En este capítulo se ofrece una introducción a los temas de la RV y la visión estereoscópica, así como las distintas formas de crear y visualizar imágenes estereoscópicas. Luego del estudio del arte se selecciona la

visión paralela como técnica de visualización estereoscópica y el visor estereoscópico cardboard como la mejor forma de solucionar el problema.

# CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

## 2.1 Introducción

En el presente capítulo se especifican las características fundamentales del módulo. Se hace referencia a las fases de exploración y planificación que plantea la metodología de desarrollo XP. Además, se muestran los requerimientos del módulo, la arquitectura y los patrones de diseño que se utilizan para la implementación del módulo a desarrollar.

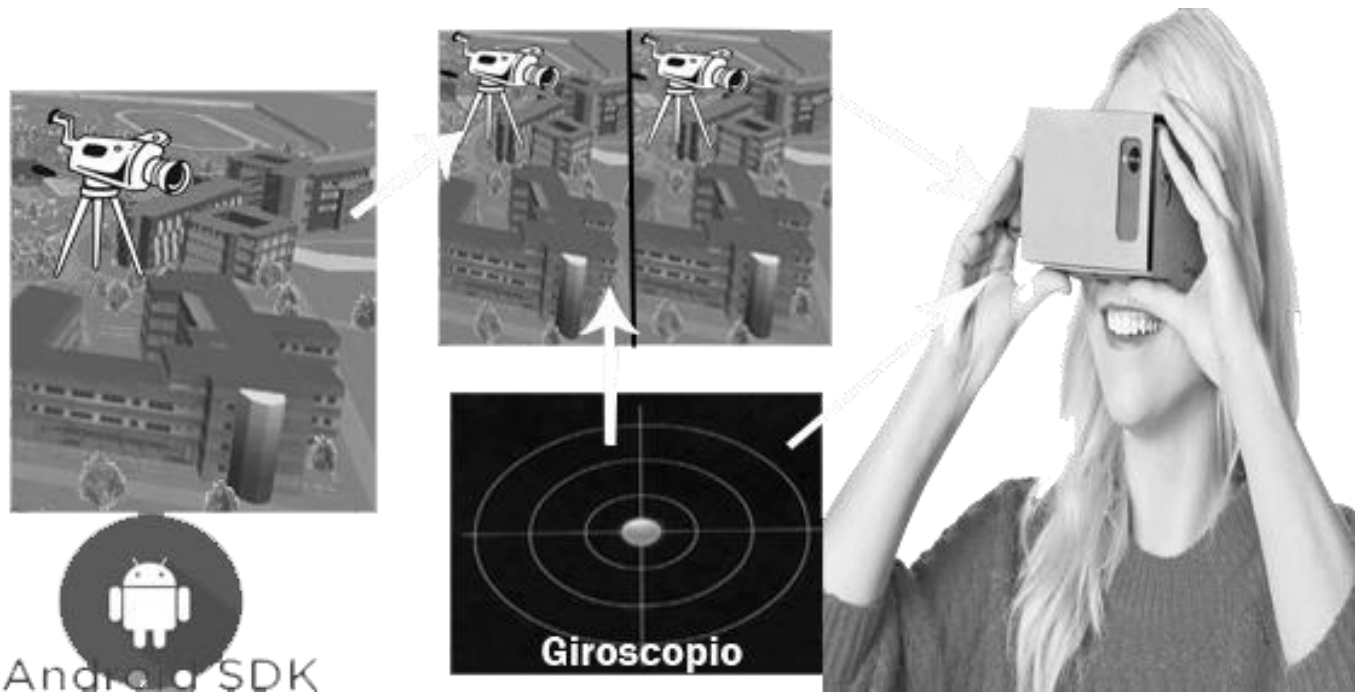
## 2.2 Propuesta de Solución

La aplicación web desarrollada por VERTEX, cuenta con todas las funcionalidades para navegar por el mapa 3D de la UCI. La cámara tiene componentes como scripts para controlar su movimiento libre, y animaciones modificables para trasladarla a un punto específico del mapa. Los métodos fundamentales son el *centerCamare()* y el *showMenu()*. La cámara, con el primer método, realiza un movimiento mediante una curva de animación hasta el edificio que el usuario desea visitar, y con el otro, regresa al punto inicial definido como centro del mapa. Las interacciones se realizan mediante botones que son activados por el ratón.

La propuesta de solución reutiliza todas las funcionalidades existentes. La cámara de la escena se divide en dos cámaras separadas a una distancia de 6cm aproximadamente. Estas cámaras muestran en una pantalla las dos imágenes captadas. El movimiento de cámara depende únicamente de los giros de la cabeza del usuario que son captados por el sensor de giroscopio del dispositivo móvil.

Los menús toman una posición estática en la escena para que el usuario, con un puntero guiado por su mirada, pueda seleccionar la acción que desea realizar. Cada vez que el puntero cambia su apariencia indica al usuario que puede interactuar.

La Figura 7 representa las modificaciones que realiza el módulo a la aplicación para llegar al usuario mediante el visor estereoscópico Cardboard.



**Figura 7: Propuesta de solución**

Los métodos fundamentales del módulo son *RecivedMessenge()* que recibe un mensaje con el menú de edificios que debe mostrar y el que debe ocultar, y el *Update()* de la clase *RayCasttest* que recibe toda la información de las colisiones del puntero y la renvía al método encargado de ejecutar la respuesta adecuada para cada situación.

A continuación, en la Figura 8 se ilustra cómo quedó definida la estructura funcional del módulo.



**Figura 8: Estructura funcional del sistema**

En la Tabla 3 se muestra una breve descripción de cada fase por la que pasa el módulo.

**Tabla 3: Estructura funcional del sistema**

No	Estado	Descripción
1	Inicialización del sistema	En esta fase la aplicación carga todos los objetos del mapa de la UCI y se prepara para mostrar al usuario todo lo que verá en un formato estereoscópico.
2	Doblar las imágenes de la cámara	Toda la información captada por la cámara debe ser doblada y redimensionada captando la misma imagen con dos cámaras, simulando los ojos humanos.

3	Mostrar imágenes en formato estereoscópico	Las 2 imágenes son mostradas con una separación aproximada de 6cm para ser perfectamente vista con un dispositivo estereoscópico.
4	Selección del tipo de edificio que se quiere visitar	En esta fase el módulo muestra un menú en forma de listado con todos los edificios de un tipo entrado por el usuario.
5	Selección del edificio dentro de los tipos seleccionados	En esta fase el módulo transporta la cámara hacia un edificio seleccionado por el usuario.
6	Navegación en el edificio seleccionado	En esta fase el módulo muestra un menú compuesto por botones para que el usuario sea capaz de trasladarse por las áreas del edificio seleccionado.

## 2.3 Requisitos no funcionales

Las aplicaciones están regidas por un grupo de restricciones que aseguran el correcto funcionamiento del software. Son conocidos como requisitos no funcionales o atributos de calidad, y son utilizados para establecer características de funcionamiento. En la investigación se realizó una captura de requisitos no funcionales, se muestran a continuación las reglas de diseño, de hardware, y de software. (18)

### 2.3.1 Captura de Requisitos no Funcionales

#### Restricciones en el Diseño e Implementación:

- Lenguaje de programación C#
- IDE de desarrollo: Visual Studio 2012
- Bibliotecas de Clases: Unity-walkthrough—master

#### Restricciones de Hardware:

- Sensor giroscopio
- Arm cortex quad-core 1Ghz
- 2 Gygabytes RAM

#### Restricciones de Software

- Sistema Operativo Android 4.4 (Kit Kat)

## 2.4 Fase de exploración

Las HU en esta fase son una selección de los clientes para la primera entrega. En cada una de ellas se especifica una de las funcionalidades que el programa tendrá. Conjuntamente el equipo de desarrollo se familiarizará con las herramientas, la tecnología y las prácticas a ser utilizadas durante el proyecto. Puede ser usado un prototipo para realizar pruebas de algunas nuevas tecnologías y explorar algunos aspectos de la arquitectura a implementar. Esta fase puede durar unas pocas semanas o extenderse de la adaptación del equipo de desarrollo.

### 2.4.1 Historias de usuario

Para el desarrollo de la investigación y el cumplimiento de su objetivo se identificaron las siguientes Historias de Usuario:

**Tabla 4: Historia de usuario 1**

Historia de Usuario	
Número:1	Usuario: Cliente
Nombre: Doblar las imágenes de la cámara.	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Mario Jorge Ramírez Pérez	
Descripción: Las imágenes captadas por la cámara de la escena deben ser procesadas, para adaptarlas a un formato estereoscópico.	
Observaciones: Con el correcto desarrollo de esta funcionalidad, se garantiza que las imágenes estén preparadas para llevar toda la aplicación a una visión estereoscópica.	

**Tabla 5: Historia de usuario 2**

Historia de Usuario	
Número:2	Usuario: Cliente
Nombre: Mostrar imágenes en formato estereoscópico.	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Mario Jorge Ramírez Pérez	



Descripción: Ya con las imágenes procesadas y dobladas son mostradas con una ligera diferencia una de otra y con una separación de 6cm aproximadamente, puede sentirse la inmersión con el visor *Cardboard*.

Observaciones: Con el correcto desarrollo de esta funcionalidad, se garantiza que las vistas de la aplicación ganen en profundidad y en sensación de inmersión.

**Tabla 6: Historia de usuario 3**

Historia de Usuario	
Número:3	Usuario: Cliente
Nombre: Mover la cámara mediante el giroscopio.	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Mario Jorge Ramírez Pérez	
Descripción: El módulo debe ser capaz de interactuar con el usuario de una forma atractiva y sencilla.	
Observaciones: Con el correcto desarrollo de esta funcionalidad, se garantiza un completo cambio de la experiencia de usuario en la aplicación. Puesto que solamente el usuario tendrá que mover la cabeza y visualizar la opción que desea realizar y esta se efectuará luego de un pequeño tiempo de confirmación.	

**Tabla 7: Historia de usuario 4**

Historia de Usuario	
Número:4	Usuario: Cliente
Nombre: Crear un puntero de selección.	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Mario Jorge Ramírez Pérez	
Descripción: Se implementará un puntero, que permitirá que el usuario interactúe con los objetos de la aplicación.	
Observaciones: Con el correcto desarrollo de esta funcionalidad, se garantiza una interactividad entre el usuario y la aplicación completamente diferente a la ya existente.	

**Tabla 8: Historia de usuario 5**

Historia de Usuario	
Número:5	Usuario: Cliente
Nombre: Adaptar selección del tipo de edificio que se quiere visitar para una vista estereoscópica.	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Mario Jorge Ramírez Pérez	
Descripción: El menú de selección de tipo de edificio debe ser modificado y adaptado para mostrar todas las opciones de edificios a visitar que se brinda en la aplicación.	
Observaciones: Con el correcto desarrollo de esta funcionalidad, se garantiza que la primera forma de interacción del usuario con la aplicación, se lo mas intuitiva y atractiva posible.	

**Tabla 9: Historia de usuario 6**

Historia de Usuario	
Número:6	Usuario: Cliente
Nombre: Adaptar selección del edificio que se quiere visitar dentro del tipo seleccionado para una vista estereoscópica.	
Prioridad del negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Mario Jorge Ramírez Pérez	
Descripción: El menú de selección de edificio, deben ser modificados y adaptados para ser seleccionados mediante un puntero dirigido por los movimientos de la cabeza.	
Observaciones: Con el correcto desarrollo de esta funcionalidad, se garantiza una fácil accesibilidad a las funcionalidades que brinda la aplicación.	

**Tabla 10: Historia de usuario 7**

Historia de Usuario	
Número:7	Usuario: Cliente
Nombre: Adaptar la navegación alrededor del edificio y el regreso al menú de selección.	
Prioridad del negocio: Media	Riesgo en desarrollo: Medio

Puntos estimados: 2	Iteración asignada: 4
Programador responsable: Mario Jorge Ramírez Pérez	
Descripción: Los menús de navegación de edificio, deben ser modificados y adaptados para facilitar que el usuario pueda interactuar con ellos.	
Observaciones: Con el correcto desarrollo de esta funcionalidad, una mayor inmersión del usuario en la aplicación.	

## 2.5 Fase de Planificación

En esta fase se fijan todas las prioridades de cada una de las HU y se establece el contenido de la primera entrega. Se realiza una estimación del esfuerzo exigido por cada HU y se establece un cronograma. La duración del calendario para la entrega de la primera liberación no suele exceder los dos meses. Y la planificación en sí no toma más de dos días.

### 2.5.1 Estimación de esfuerzo por Historias del usuario

Es responsabilidad de los programadores estimar el esfuerzo necesario para dar cumplimiento a cada HU siguiendo la prioridad establecida por el cliente. Para dicha estimación se toma como medida el punto de estimación, el cual equivale a una semana de trabajo ideal de programación, donde los programadores trabajan el tiempo planeado sin ningún tipo de interrupción.

Para el desarrollo de la aplicación se realizó una estimación del esfuerzo para cada una de las HU identificadas, alcanzando los resultados que se muestran en la Tabla 11.

**Tabla 11: Estimación de esfuerzo**

Historia de Usuario	Puntos de Estimación
Doblar las imágenes de la cámara	1
Mostrar imágenes en formato estereoscópico	1
Mover la cámara mediante el giroscopio (sensor del dispositivo móvil)	1
Crear un puntero de selección.	2
Adaptar selección del tipo de edificio que se quiere visitar para una vista estereoscópica.	1
Adaptar selección el edificio que se quiere visitar dentro del tipo seleccionado para una vista estereoscópica.	1

Adaptar la navegación alrededor del edificio y el regreso al menú de selección.	1 semana
---	----------

### 2.5.2 Plan de iteraciones

Una vez definidas las HU y estimado el esfuerzo propuesto para la realización de cada una de ellas, el desarrollo del sistema se distribuyó en 4 iteraciones, las cuales se describen a continuación de manera más detallada:

#### 1. Primera Iteración

Tuvo como objetivo la implementación de las siguientes HU:

- HU 1. Doblar las imágenes de la cámara
- HU 2. Mostrar imágenes en formato estereoscópico

Al concluir esta iteración, se preparó al sistema para el desarrollo de la siguiente iteración. Una vez terminada esta iteración se contó con la primera versión de prueba del sistema.

#### 2. Segunda Iteración

En esta iteración se llevó a cabo la implementación de las siguientes HU:

- HU 3. Mover la cámara mediante el giroscopio.
- HU 4. Crear un puntero de selección.

De igual manera, al concluir esta iteración, se procedió al desarrollo de la siguiente iteración. El resultado alcanzado en esta iteración proporcionó una versión de prueba del sistema.

#### 3. Tercera Iteración

Tiene como objetivo la implementación de las siguientes HU:

- HU 5. Adaptar selección del tipo de edificio que se quiere visitar para una vista estereoscópica.
- HU 6. Adaptar selección el edificio que se quiere visitar dentro del tipo seleccionado para una vista estereoscópica.

Al concluir esta iteración se resolvió gran parte del problema que fue planteado en el momento que surgió la necesidad de implementar la aplicación. Es válido señalar que el desarrollo de Iteración 4 depende fuertemente del desarrollo de las HU anteriores. Hasta aquí todo es un proceso secuencial, en el que la salida obtenida en cada HU es utilizada en la siguiente. Con el cumplimiento de esta iteración se contó con una tercera versión de prueba del módulo.

#### 4. Cuarta Iteración

En esta última iteración se llevó a cabo la implementación de la siguiente HU

- HU 7. Adaptar la navegación alrededor del edificio y el regreso al menú de selección.

Esta HU es integrada con el resultado de las iteraciones antes descritas y como beneficio de esta integración se obtuvo la aplicación de VR, basada en la vista estereoscópica usando la tecnología *Cardboard*.

### 2.5.3 Plan de duración de las iteraciones

**Tabla 12: Plan de duración**

No Iteración	Historia de Usuario	Duración Total de Iteraciones
<b>Iteración 1</b>	-Doblar las imágenes de la cámara. -Mostrar imagen en formato estereoscópico.	2 semanas
<b>Iteración 2</b>	-Mover la cámara mediante el giroscopio -Crear un puntero de selección.	3 semanas
<b>Iteración 3</b>	-Adaptar selección del tipo de edificio que se quiere visitar para una vista estereoscópica. -Adaptar selección el edificio que se quiere visitar dentro del tipo seleccionado para una vista estereoscópica.	2 semanas
<b>Iteración 4</b>	-Adaptar la navegación alrededor del edificio y el regreso al menú de selección.	1 semana

### 2.5.4 Plan de entrega

Una vez definido el plan de iteraciones, se pasa a elaborar el plan de entrega. Este tiene como principal objetivo dejar plasmado el orden en que se realizarán las diferentes entregas intermedias y la entrega final de la aplicación.

**Tabla 13: Plan de entrega**

No. Iteración	Duración	Fecha Inicio	Fecha Final
<b>Iteración 1</b>	2 semanas	12/2/2018	23/2/2018

<b>Iteración 2</b>	3 semanas	26/2/2018	9/3/2018
<b>Iteración 3</b>	2 semanas	12/3/2018	23/3/2018
<b>Iteración 4</b>	1 semana	26/3/2018	30/3/2018

Tabla 14: Plan de entrega

## 2.6 Diseño de la solución propuesta

### 2.6.1 Arquitectura de Software

El estilo arquitectónico que se le aplicó al sistema fue el conocido estilo en capas específicamente en tres capas el cual se muestra en la Figura 9:

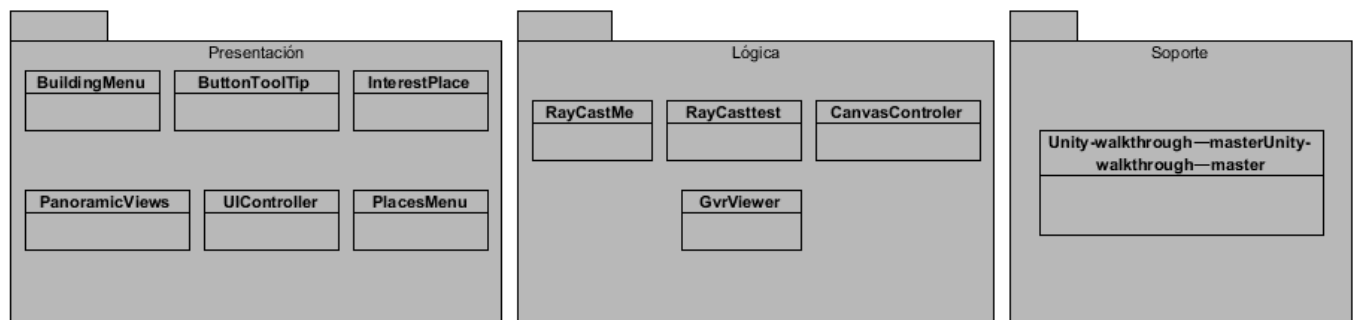


Figura 9: Diseño arquitectónico del sistema

- **Presentación:** En esta capa se ubican las interfaces del sistema que muestran al usuario todas las formas de interactuar con la aplicación, como botones, edificios y menús de navegación.
- **Lógica:** En esta capa se encuentran aquellas clases encargadas de realizar las validaciones correspondientes a las entradas realizadas por el usuario. Se realiza todo el proceso de doblaje y *render* de las imágenes captadas por la cámara de la escena. Para que la capa Lógica pueda realizar exitosamente todas sus funciones, esta se apoya en la capa de Soporte.
- **Soporte:** En esta capa se encuentran los elementos necesarios para que la capa de Lógica desarrolle correctamente todas sus funciones. Esta capa enmarca elementos como bibliotecas (*Unity-walkthrough—master*) y datos constantes.

### 2.6.2 Patrones de diseño

Para el diseño de las clases se utilizó el conjunto de patrones de diseño *GRASP* (*General responsibility assignment software patterns*) y *GOF* (*Gang of Four*), que intervienen en la asignación de responsabilidades, estructura y comportamiento respectivamente. A continuación, se indicarán aquellos que fueron aplicados.

### 2.6.3 Patrones GRASP

El *Patrón Experto* se utilizó más que cualquier otro al asignar responsabilidades; pues es un principio básico que suele utilizarse en el diseño orientado a objetos. Dio origen a diseños donde el objeto de software realiza las operaciones que normalmente se aplican al concepto que representa, por lo que ofrece una analogía con el mundo real. Permitió asignarles las responsabilidades necesarias a desarrollar a cuyas clases contienen la información para realizarlas. En este caso cada clase se diseñó según su función en el proceso (21).

Otro patrón que se pone de manifiesto es el Bajo Acoplamiento, que trata de establecer la menor dependencia posible entre las clases, de modo que sea más adaptable a cambios futuros. En este patrón se hace necesario ya que, al estar desarrollando sobre una aplicación con sus propias clases y su propia ingeniería, es más factible desarrollar clases independientes para que los cambios en ellas no afecten al resto de la aplicación. Una de las principales Clases donde se pone de manifiesto es en “RayCastMe”, clase que se encarga de tomar la decisión correcta cuando identifica que está siendo casteado por el puntero de selección.

### 2.6.4 Patrones GOF

De los patrones GOF se hace uso del *Singleton*, es necesario que todas las instancias de las clases sean únicas y puedan ser accedidas desde cualquier parte de la aplicación. Las clases ya existentes en la aplicación original son instanciadas en las nuevas para una mejor comunicación. Un ejemplo de su uso puede encontrarse en la Clase “*RayCasttest*”, que se encarga de controlar las selecciones hechas por el puntero a los objetos con los que el usuario puede interactuar (21).

### 2.6.5 Tarjetas CRC

A continuación, se muestran las tarjetas CRC, las cuales son fichas, una por cada clase, en las que se escriben brevemente las responsabilidades de cada una e informa además su relación respectiva con otras clases, las cuales garantizan poder llevar a cabo dichas responsabilidades.

**Tabla 15: Tarjeta CRC 1**

Tarjeta CRC

<b>Clase: Gvr Viewer</b>	
<b>Responsabilidad</b>	Colaboración
<b>Es responsable de:</b> <ul style="list-style-type: none"> <li>• Consultar el dispositivo para ver los parámetros</li> <li>• Recuperar los últimos datos de seguimiento de la cabeza</li> <li>• Proporcionar la escena renderizada al dispositivo para corregir la distorsión.</li> </ul>	StereoController Camera

**Tabla 16: Tarjeta CRC 2**

Tarjeta CRC	
<b>Clase: RayCastMe</b>	
<b>Responsabilidad</b>	Colaboración
<b>Es responsable de:</b> <ul style="list-style-type: none"> <li>• Recibir los mensajes que envía el puntero de selección.</li> <li>• Identificar y validar las peticiones del usuario y da paso para que la clase encargada de ejecutarla entre en acción.</li> </ul>	

**Tabla 17: Tarjeta CRC 3**

Tarjeta CRC	
<b>Clase: RayCasttest</b>	
<b>Responsabilidad</b>	Colaboración
<b>Es responsable de:</b> <ul style="list-style-type: none"> <li>• Instanciar un puntero de selección que sigue todos los movimientos del usuario.</li> <li>• Ser el canal mediante el cual el usuario puede interactuar con la aplicación.</li> </ul>	

**Tabla 18: Tarjeta CRC 4**

Tarjeta CRC	
<b>Clase: CanvasControler</b>	
<b>Responsabilidad</b>	Colaboración
<b>Es responsable de:</b> <ul style="list-style-type: none"> <li>• Que los componentes visuales del menú de navegación del edificio persigan constantemente la mirada del usuario.</li> <li>• Estos componentes solo deben ser visibles cuando el usuario se encuentre en el edificio a visitar.</li> </ul>	

## 2.7 Conclusiones Parciales

La implementación de esta lógica permitió demostrar que la solución propuesta resuelve el problema de aumentar la inmersión en el Paseo Virtual UCI WEB. Se definió una estructura para dicha aplicación que contiene características y funcionalidades que permiten la combinación de algoritmos de detección y



extracción de imágenes en formato estereoscópico. Con esto se garantizó la interacción con la aplicación usando la tecnología *Cardboard*.

# CAPÍTULO 3: EVALUACIÓN DE LA SOLUCIÓN PROPUESTA

## 3.1 Introducción

El presente capítulo está compuesto por una exposición de todos los detalles de la implementación del módulo. Así como la definición de las tareas ingenieriles que rigieron el desarrollo de la solución y a las pruebas realizadas al módulo con sus principales resultados obtenidos, demostrando la eficiencia de la solución dada al problema.

## 3.2 Estándar de Codificación

Los programadores deben seguir estrictamente un grupo de reglas y condiciones conocidas como el estándar de codificación. En un equipo de programadores todos renuncian a su forma de programar particular y se ajustan a un estándar para el código. De esta forma al final de toda la implementación parece hecho por una misma persona. El ejemplo de esta buena práctica de desarrollo de software es de gran importancia y se traduce en una mayor calidad de desarrollo (22).

El estándar de codificación conduce a una mayor coherencia entre el código personal y el de los compañeros del equipo, y esto a su vez permite generar un código más fácil de entender que facilita su desarrollo y mantenimiento. Esto reduce el costo total de las aplicaciones a crear.

Para el desarrollo de este módulo, se estableció un estándar de codificación propio. A continuación, se muestran las pautas que lo conforman:

- Los nombres de las clases, métodos y variables serán escritos en idioma inglés.
- Los nombres de las clases y estructuras serán escritos con letra inicial mayúscula.
- En el caso de que las clases y estructuras tengan un nombre compuesto, cada palabra comenzará con mayúscula, tal y como propone el estándar *Camel Case*.
- Los nombres de las variables y métodos se escribirán en minúscula.
- En caso de que los métodos y variables sean compuestos, la palabra siguiente comenzará con mayúscula.
- Los comentarios de implementación se delimitarán entre `/*...*/` o se utilizará la variante `//`.
- Los comentarios de documentación se delimitarán por `/**...*/`.

En la Figura 10 se aprecia un correcto uso del estándar de codificación.

```

public class RayCastMe : MonoBehaviour {
    private string hitButton;
    private string lastButton;
    public GameObject imgSelect;
    void Start()
    {
        hitButton = "";
        lastButton = "";
    }
    private void ReceivedMessage()
    {
        lastButton = FindObjectOfType<raycasttest>().getlastButtons();

        LastMenu(lastButton);
        hitButton = FindObjectOfType<raycasttest>().gethitButtons();
        if(hitButton!=lastButton)
        {
            LastMenu(lastButton);
            if (hitButton=="BuildingsBtnCollider")
            {
                FindObjectOfType<PlacesMenu>().ToggleBuildingsMenu(true);
            }
            if (hitButton=="SculptureBtnCollider")
            {
                FindObjectOfType<PlacesMenu>().ToggleSculpturesMenu(true);
            }
            if (hitButton=="SportsBtnCollider")
            {
                FindObjectOfType<PlacesMenu>().ToggleSportsMenu(true);
            }
        }
    }
}

```

**Figura 10: Fragmento de código**

### 3.3 Fase de Producción

La fase de producción requiere pruebas extras y chequeos de la ejecución del sistema antes de que sea entregado al cliente. En esta fase, se pueden encontrar nuevos cambios y se decide si serán incluidos en la entrega actual. Durante esta fase, las iteraciones pueden necesitar ser recortadas de tres semanas a una semana. Después que la primera entrega es producida para el uso del cliente, el proyecto XP debe mantener el sistema en producción corriendo.

En el capítulo anterior se realizó una descripción y distribución por iteraciones de todas las HU a desarrollar, pero esta información por sí sola no brinda suficientes elementos que permitan su análisis y desarrollo. Es por este motivo que para cada iteración se establecen las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. Consecuentemente se descomponen todas las UH en tareas de programación o ingeniería, que pueden ser redactadas en un lenguaje técnico debido a que son para uso estricto de los programadores, donde el usuario no tiene que

necesariamente comprenderlas. A continuación, se muestran las tareas de ingeniería que quedaron definidas por cada iteración e HU correspondiente.

**Tabla 19: Tareas de Iteración 1**

Tareas de la Iteración 1	
Historia de Usuario	Tareas
Doblar las imágenes de la cámara	Doblar las imágenes captadas por la cámara para renderizarlas mediante un script.
Mostrar imágenes en formato estereoscópico	Proporcionar la escena renderizada al dispositivo para corregir la distorsión.

**Tabla 20: Tareas de ingeniería 1**

Tareas de Ingeniería	
No. de la tarea: 1	<b>No. HU: 1</b>
Nombre de la Tarea: Doblar las imágenes captadas por la cámara para renderizarlas mediante un script.	
Tipo de tarea: Desarrollo	Puntos estimados: 7 días
Fecha de Inicio: 12/02/2018	Fecha final: 16/02/2018
Programador Responsable: Mario Jorge Ramírez Pérez	
Descripción: Se implementará un script en C# usando la biblioteca Google VR, utilizando las imágenes captadas por la cámara de la escena.	

Las restantes tareas de ingeniería se encuentran en el anexo 1.

### 3.3.1 Pruebas

Uno de los pilares de la metodología XP es el uso de las pruebas para comprobar el funcionamiento de los códigos implementados. Esto permite aumentar la calidad de los sistemas, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

La metodología ágil XP divide las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación. Las pruebas unitarias son desarrolladas por los programadores y se encargan de verificar el código automáticamente y las pruebas de aceptación están destinadas a verificar que al final de cada iteración las Historias de Usuario cumplen con la funcionalidad asignada y satisfacen las necesidades del cliente. Las pruebas de

aceptación son más importantes que las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente, por esto el cliente es la persona adecuada para diseñar las pruebas de aceptación.

### 3.3.2 Entorno de Prueba

Las pruebas fueron realizadas en un dispositivo móvil con las siguientes propiedades:

- Pantalla de 5 pulgadas, de alta resolución(HD).
- Versión Android 4.4 Kit Kat
- Procesador Qualcomm Snapdragon 600 Quad-Core a 1.9GHz
- 2 Gygabytes de RAM

### 3.3.3 Pruebas de aceptación

Las pruebas de aceptación fueron creadas en base a las HU, en cada ciclo de la iteración del desarrollo. El cliente especificó uno o diversos escenarios para comprobar que una historia de usuario fue correctamente implementada. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. A continuación, se muestran las pruebas más significativas realizadas a cada una de las Historias de Usuario.

**Tabla 21: Caso de prueba de aceptación 1**

Caso de Prueba de Aceptación	
Código: HU1_P1	<b>Historia de Usuario: 1</b>
Nombre: Doblar las imágenes de la cámara	
Condiciones de ejecución: La escena debe tener una instancia única del objeto GvrViewer y la cámara.	
Pasos de Ejecución:	
1: Las imágenes de la cámara son tomadas.	
2: Las imágenes son dobladas y renderizadas.	
Resultados esperados:	
1: Las imágenes quedan listas para el <i>render</i>	
Evaluación de la prueba: Satisfactoria	

**Tabla 22: Caso de prueba de aceptación 2**

Caso de Prueba de Aceptación	
Código: HU2_P1	<b>Historia de Usuario: 2</b>
Nombre: Mostrar imágenes en formato estereoscópico.	
Condiciones de ejecución:	

Pasos de Ejecución:
1: Se muestran las imágenes ya procesadas y dobladas.
Resultados esperados:
1: Las imágenes son mostradas ya renderizadas y en formato estereoscópico.
Evaluación de la prueba: Satisfactoria

**Tabla 23: Caso de prueba de aceptación 3**

Caso de Prueba de Aceptación	
Código: HU3_P1	<b>Historia de Usuario: 3</b>
Nombre: Mover la cámara mediante el giroscopio	
Condiciones de ejecución: El dispositivo móvil tiene que contener un sensor giroscópico.	
Pasos de Ejecución:	
1: El usuario mueve la cabeza buscando el objeto con el que desea interactuar.	
Resultados esperados:	
1: La cámara se mueve de la misma forma que lo hace la cabeza del usuario.	
Evaluación de la prueba: Satisfactoria	

**Tabla 24: Caso de prueba de aceptación 4**

Caso de Prueba de Aceptación	
Código: HU4_P1	<b>Historia de Usuario: 4</b>
Nombre: Crear un puntero de selección.	
Condiciones de ejecución: El usuario debe fijar un objeto que tenga interacción.	
Pasos de Ejecución:	
1: El usuario señala un objeto con el que puede interactuar.	
2: El usuario mantiene la selección durante el tiempo de confirmación.	
Resultados esperados:	
1: Si el usuario señala un objeto para interactuar, es mostrado un contador de 3 segundos para confirmar.	
2: Si el usuario no mantiene la selección durante todo el tiempo de confirmación el contador desaparece.	
3: Si se cumplen correctamente el paso 1 y 2 se realizará la acción que corresponde al objeto seleccionado.	
Evaluación de la prueba: Satisfactoria	

**Tabla 25: Caso de prueba de aceptación 5**

Caso de Prueba de Aceptación	
Código: HU5_P1	<b>Historia de Usuario: 5</b>
Nombre: Adaptar selección del tipo de edificio que se quiere visitar para una vista estereoscópica.	
Condiciones de ejecución: El usuario debe seleccionar el tipo de edificio que quiere visitar.	
Pasos de Ejecución: 1: El usuario selecciona el botón referente al tipo de edificio que desea seleccionar. 2: El usuario mantiene la selección durante el tiempo de confirmación.	
Resultados esperados: 1: Si el usuario señala el botón para interactuar, es mostrado un contador de 3 segundos para confirmar y el tipo de edificio que está en su selección. 2: Si el usuario no mantiene la selección durante todo el tiempo de confirmación el contador desaparece. 3: Si se cumplen correctamente el paso 1 y 2 se mostrará un menú en forma de espiral con todos los edificios del tipo seleccionado.	
Evaluación de la prueba: Satisfactoria	

**Tabla 26: Caso de prueba de aceptación 6**

Caso de Prueba de Aceptación	
Código: HU6_P1	<b>Historia de Usuario: 6</b>
Nombre: Adaptar selección del edificio que se quiere visitar dentro del tipo seleccionado para una vista estereoscópica.	
Condiciones de ejecución: El usuario debe seleccionar el tipo de edificio que quiere visitar.	
Pasos de Ejecución: 1: El usuario selecciona el botón referente al edificio que desea seleccionar. 2: El usuario mantiene la selección durante el tiempo de confirmación.	
Resultados esperados: 1: Si el usuario señala el botón para interactuar, es mostrado un contador de 3 segundos para confirmar. 2: Si el usuario no mantiene la selección durante todo el tiempo de confirmación el contador desaparece. 3: Si se cumplen correctamente el paso 1 y 2 la cámara de la escena será trasladada al edificio seleccionado y se mostrará un menú de navegación de edificios.	
Evaluación de la prueba: Satisfactoria	

**Tabla 27: Caso de prueba de aceptación 7**

Caso de Prueba de Aceptación	
<b>Código:</b> HU7_P1	Historia de Usuario: 7
<b>Nombre:</b> Adaptar la navegación alrededor del edificio y el regreso al menú de selección.	
<b>Condiciones de ejecución:</b> El usuario debe seleccionar el edificio que quiere visitar.	
<b>Pasos de Ejecución:</b>	
1: El usuario selecciona el botón referente a la acción sobre el edificio que desea realizar.	
2: El usuario mantiene la selección durante el tiempo de confirmación.	
<b>Resultados esperados:</b>	
1: Si el usuario señala el botón para interactuar, es mostrado un contador de 3 segundos para confirmar.	
2: Si el usuario no mantiene la selección durante todo el tiempo de confirmación el contador desaparece.	
3: Si se cumplen correctamente el paso 1 y 2 se realizará la acción que corresponde al botón seleccionado.	
<b>Evaluación de la prueba:</b> Satisfactoria	

En la revisión de las funcionalidades se realizaron cuatro iteraciones de pruebas, detectándose en la primera de ellas un total de dos no conformidades, en la segunda y tercera se detectaron tres no conformidades en cada una. La mayoría de estas no conformidades fueron errores de validación, los cuales fueron solucionados satisfactoriamente. En la cuarta iteración no se encontraron no conformidades alcanzando resultados satisfactorios.

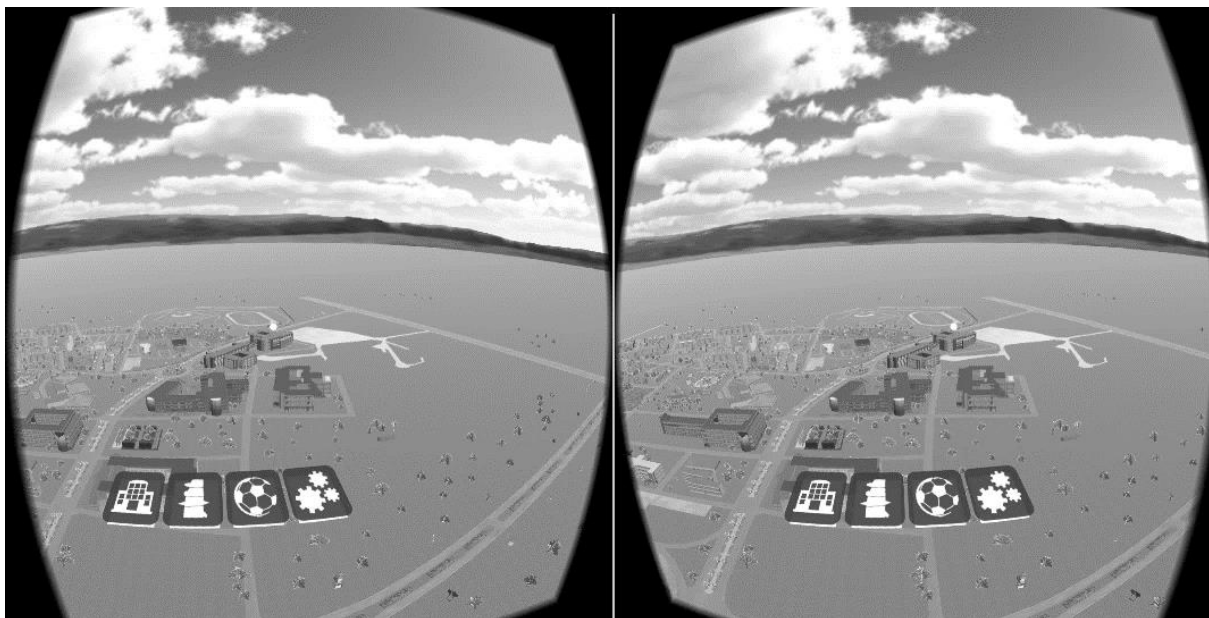
### **3.4 Resultados Obtenidos**

La realización de la presente investigación, permitió implementar un módulo de visualización estereoscópica del Paseo Virtual UCI, para teléfonos inteligentes con sistema operativo Android. Se incorpora el trabajo con el sensor de giroscopio para garantizar el seguimiento de la posición del usuario que utiliza el visor de estereoscopia *Cardboard*.

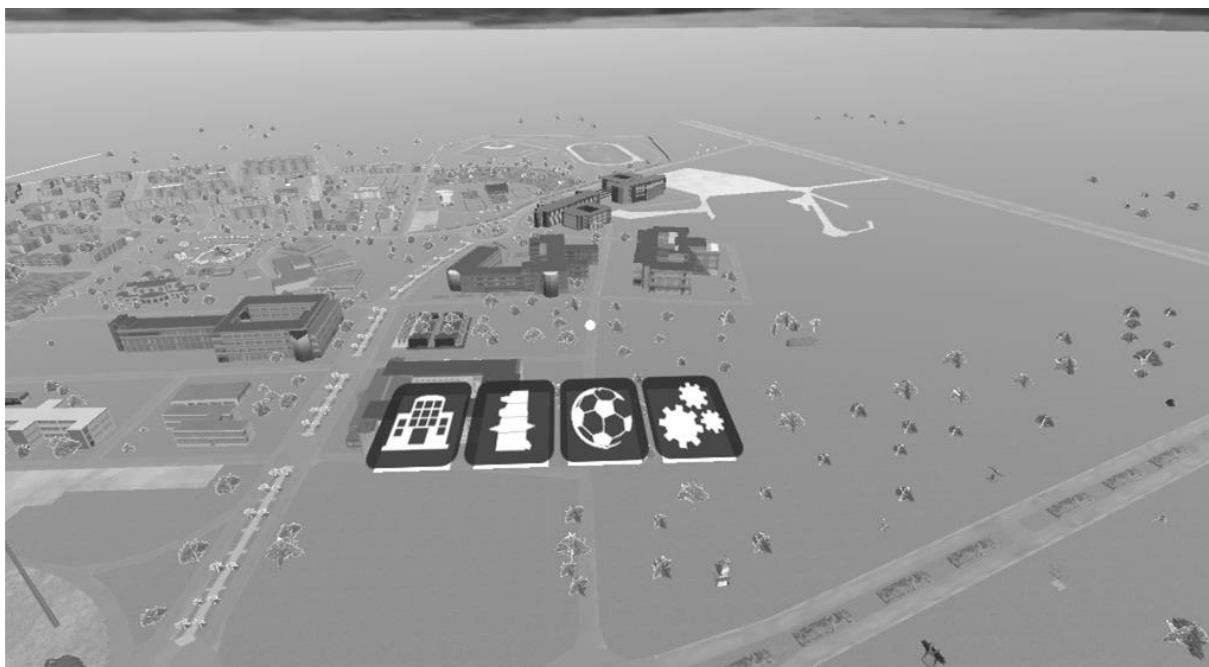
La incorporación del módulo a la aplicación en su versión WEB, generó un Paseo Virtual UCI con una mayor inmersión y experiencia de usuario dando cumplimiento al objetivo planteado y resolviendo el problema de la investigación.



En la Figura 11 se muestra una imagen donde se puede apreciar un formato estereoscópico de la primera vista que recibe el usuario del Paseo Virtual UCI. A partir de esta escena comienza la navegación por las distintas estructuras y lugares de la universidad. Este formato y la incorporación del visor estereoscópico *Cardboard*, dieron forma a una imagen con sensación de inmersión como lo muestra la Figura 12:



**Figura 11: Vista estereoscópica de la escena**



**Figura 12: Vista de usuario con el visor Cardboard**

### **3.4.5 Encuesta de Aceptación**

Para garantizar el cumplimiento del objetivo de la investigación se realiza una encuesta a una muestra de 20 personas de la UCI con el formato que se muestra en Figura 13:

La inmersión en una aplicación es la sensación que se le brinda al usuario de sentirse enajenado del mundo real y más participe del mundo virtual.

1-) ¿Según lo anteriormente planteado crees que el Paseo Virtual UCI Cardboard tiene mayor inmersión que el Paseo Virtual UCI WEB? Marque con una x su respuesta.

Si                       No sé                       No

2-) ¿Tuvo una mejor experiencia en el Paseo Virtual UCI Cardboard que en la versión WEB? Marque con una x su respuesta.

Si                       NO sé                       No

3-) ¿Le gusta el Paseo Virtual UCI Cardboard? Marque con una x su respuesta.

Me gusta mucho    No me gusta tanto    Me da lo mismo    No me gusta nada  
 No sé

4-) ¿Qué es lo que más te gusta del Paseo Virtual UCI Cardboard?

5-) ¿Qué es lo que más te gusta del Paseo Virtual UCI Cardboard?

### Figura 13: Encuesta a usuarios sobre producto final

Para el análisis de la satisfacción del usuario con respecto a la aplicación resultante, se utiliza la técnica ladov y se relacionan las 3 preguntas con respuesta cerrada mediante el Cuadro Lógico ladov que se muestra a continuación (23):

**Tabla 28: Cuadro Lógico de ladov**

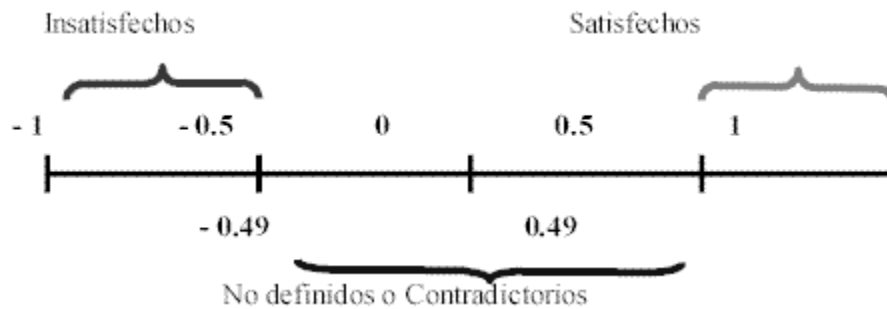
¿Según lo anteriormente planteado crees que el Paseo Virtual UCI Cardboard tiene mayor inmersión que el Paseo Virtual UCI WEB?									
	Si			No sé			No		
¿Le gusta el Paseo Virtual UCI Cardboard?	¿Tuvo una mejor experiencia en el Paseo Virtual UCI Cardboard que en la versión WEB?								
	Si	No sé	No	Si	No sé	No	Si	No sé	No
Me gustó mucho	1	2	6	2	2	6	6	6	6
No me gustó tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	4	4	3	4	4
No me gustó nada	6	6	6	6	4	4	6	4	5
No sé	2	3	6	3	3	3	6	3	4

El número resultante de la interrelación de las 3 preguntas indica la posición de la persona en la escala de satisfacción siguiente:

1. Clara satisfacción.
2. Más satisfecho que insatisfecho.

3. No define.
4. Más insatisfecho que satisfecho.
5. Clara insatisfacción.
6. Contradictoria.

Quedando una distribución de 9 personas con clara satisfacción, 5 más satisfecho que insatisfecho, 5 que no definen y 1 más insatisfecho que satisfecho. Dando como resultado un valor de satisfacción grupal de 0.55, que según la Figura 14 representa que los usuarios están satisfechos.



**Figura 14: Eje de satisfacción (Técnica ladov)**

### 3.5 Conclusiones Parciales

Con la finalización de este capítulo se logró definir el estándar de codificación, para una mayor comprensión del código y facilitando su reutilización. Las pruebas de aceptación permitieron guiar el desarrollo del sistema hasta obtener los resultados esperados. Se realiza un cuestionario, aplicándolo a 20 personas y evidenciando que la incorporación del módulo aumenta la aceptación de la aplicación.

## **CONCLUSIONES**

. En la presente investigación se concluye lo siguiente:

- Con la incorporación de vistas estereoscópicas y controles inteligentes, puede lograrse una mayor inmersión y experiencia de usuario en aplicaciones como los paseos virtuales.
- La tecnología Cardboard destaca por su compatibilidad con la mayoría de los teléfonos inteligentes, por su fácil acceso, bajo costo y la efectividad de su uso para aplicaciones como el Paseo Virtual UCI.

## **RECOMENDACIONES**

- Adaptar la aplicación para móviles sin giroscopio.
- Incorporar audio estéreo para una mayor inmersión.

# ANEXOS

## ANEXO 1 TAREAS DE INGENIERÍA

**Tabla 29: Tareas de ingeniería 2**

Tareas de Ingeniería	
No. de la tarea: 1	No. HU: 2
Nombre de la Tarea: Proporcionar la escena renderizada al dispositivo para corregir la distorsión.	
Tipo de tarea: Desarrollo	Puntos estimados: 7 días
Fecha de Inicio: 19/02/2018	Fecha final: 23/02/2018
Programador Responsable: Mario Jorge Ramírez Pérez	
Descripción: Se implementará un script en c# usando la biblioteca Google VR, utilizando las imágenes captadas por la cámara de la escena, procesadas y dobladas para mostrarlas al usuario en un formato estereoscópico.	

**Tabla 30: Tareas ingeniería 3**

Tareas de Ingeniería	
No. de la tarea: 1	No. HU: 3
Nombre de la Tarea: Implementación de controles inteligentes.	
Tipo de tarea: Desarrollo	Puntos estimados: 7 días
Fecha de Inicio: 26/02/2018	Fecha final: 2/03/2018
Programador Responsable: Mario Jorge Ramírez Pérez	
Descripción: Se implementará un script en c# usando la biblioteca Google VR, se tomará como datos de entrada la información del giroscopio para con estos definir el movimiento del usuario alrededor de la escena.	

**Tabla 31 : Tareas ingeniería 4**

Tareas de Ingeniería	
No. de la tarea: 1	No. HU: 4
Nombre de la Tarea: Implementación de un puntero que interactúe con la aplicación.	
Tipo de tarea: Desarrollo	Puntos estimados: 14 días
Fecha de Inicio: 5/02/2018	Fecha final: 16/03/2018
Programador Responsable: Mario Jorge Ramírez Pérez	

Descripción: Se implementará un script en c#, con un puntero encargado del seguimiento de usuario y le proporciona una forma fácil e intuitiva de interactuar con el medio virtual que le rodea.

**Tabla 32 : Tareas ingeniería 5**

Tareas de Ingeniería	
No. de la tarea: 1	No. HU: 5
Nombre de la Tarea: Activación de botones de selección de tipo de edificio por medio del puntero.	
Tipo de tarea: Desarrollo	Puntos estimados: 7 días
Fecha de Inicio: 17/03/2018	Fecha final: 23/03/2018
Programador Responsable: Mario Jorge Ramírez Pérez	
Descripción: Los botones existentes en la aplicación son activados mediante un clic, por lo que en la nueva versión tienen que ser activados por el puntero.	

**Tabla 33 : Tareas ingeniería 6**

Tareas de Ingeniería	
No. de la tarea: 1	No. HU: 6
Nombre de la Tarea: Activación de botones de selección de edificio por medio del puntero.	
Tipo de tarea: Desarrollo	Puntos estimados: 7 días
Fecha de Inicio: 26/04/2018	Fecha final: 30/04/2018
Programador Responsable: Mario Jorge Ramírez Pérez	
Descripción: Los botones existentes en la aplicación son activados mediante un clic, por lo que en la nueva versión tienen que ser activados por el puntero.	

**Tabla 34 : Tareas ingeniería 7**

Tareas de Ingeniería	
No. de la tarea: 1	No. HU: 7
Nombre de la Tarea: Activación de botones de navegación de edificio por medio del puntero	
Tipo de tarea: Desarrollo	Puntos estimados: 7 días
Fecha de Inicio: 2/04/2018	Fecha final: 6/04/2018
Programador Responsable: Mario Jorge Ramírez Pérez	

Descripción: Los botones existentes en la aplicación son activados mediante un clic, por lo que en la nueva versión tienen que ser activados por un puntero.



## REFERENCIAS BIBLIOGRÁFICAS

1. **Fabola, Adeola, Miller, Alan y Fawcett, Richard.** *Exploring the past with Google Cadboard.* 2015.
2. **MARTÍN, Eva Domínguez.** *Periodismo inmersivo: La influencia de la realidad virtual y del videojuego en los contenidos informativos.* s.l. : Editorial UOC, 2014.
3. **Matthews, Monte, Trevarrow, Bill y Matthews, Jennifer.** *A virtual tour of the guide for zebrafish users.* 2002.
4. **Garcia, Alfaro, y otros.** *Creacion de un tour virtual para promover el turismo en los sitios arqueologicos que forman parte de la ruta maya en el salvador.* 2010.
5. **Francois, baily y MccLendon, Brian.** *Virtual tour of user define paths in a geograph information system.* 2014.
6. **Sugand, Kapil.** *Training effect of a virtual reality haptics enable dynamic hip screw simulator.* 2015.
7. **Univio, Monroy.** *Diseño de una aplicacion basada en realidad virtual inmersiva.* 2016.
8. **Dominguez, Martin.** *periodismo inmersivo cómo la realidad virtual y el video juego influyen en la interfaz e interactividad del relato de actualidad.* 2015.
9. **Hernandez, Ambrosio Javier.** *Reconstruccion y visualizacion 3D del interior de la celula.*
10. **Armáis, Jorge L.** *La primera sociedad mediatica fotografía, esteroscopía .* 2016.
11. **Gonzalez, Soledad.** *Diseño y deasrrollo de un video juego educativo con tecnicas de inteligencia artificial.* 2016.
12. **Kim, Sung Lae.** *using unity 3D to facilitate mobile augmented reality game development.* 2014.
13. **EGGER, Jan, et al.** *HTC Vive MeVisLab integration via OpenVR for medical applications.* 2017.
14. **DESAI, Parth Rajesh, et al.** *A review paper on oculus rift-a virtual reality headset.* 2014.
15. **MÓL, Antônio Carlos Abreu, et al.** *The use of the virtual reality Helmet Samsung Gear vr as interaction interface of a radioactive waste repository simulator.* 2017.
16. **Viña Leon, Nestor.** *Tecnologia de la realidad virtual para el tratamiento de ambliopía en adultos.* 2015.
17. **Fernandez Salvador, Prado.** *Aplicacio de la metodologia de desarrollo de ingenieria de software extrme programming (XP) .* 2016.
18. **Selma Garcia, Alejandra.** *Desarrollo de aplicaciones para plataformas y uso de unity 3D.* 2017.
19. **Hocking, Joseph.** *Multiplataform game development in c# with unity.* 2015.

20. **Buitrón, Sandra.** *Transformación del conocimiento en el proceso de elicitación de requisitos no funcionales.* 2016.
21. **Menesses Yero, Mario Alexander.** *Editor para el lenguaje lista de instrucciones.* 2013.
22. **Network.** *Microsoft developer. revisiones de código y estándar de codificación.* 2015.
23. **LÓPEZ, A. y GONZÁLEZ, V.** *La técnica de ladov. Una aplicación para el estudio de la satisfacción de los alumnos por las clases de educación física.* 2002.
24. **Telo , Jesus.** *Patrones diseño.* 2008.
25. **Pitti, Carmen.** *El uso combinado de la exposición a realidad virtual en tratamiento de la agorafobia.* 2015.
26. **Ordoñez, H.** *Business process as a strategy to improve re-quirements elicitation in XP.* 2015.
27. **Gasca Hurtado, Gloria Piedad.** *realidad virtual como buena practica para trabajo en equipo en trabajos de ingeniería.* 2015.