

**Universidad de las Ciencias Informáticas**  
**Facultad 4**



**Título: "Propuesta de estándar para la creación de Paseos  
Virtuales en la web"**

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

**Autor:** Damian Hernández Farramola

**Tutores:** Ing. Jandy Miguel Gómez Rodríguez

Ing. Luis Guillermo Silva Rojas

**La Habana, Cuba**  
**2017**

## PENSAMIENTO

“Tu tiempo es limitado, de modo que no lo malgastes viviendo la vida de alguien distinto. No quedes atrapado en el dogma, que es vivir como otros piensan que deberías vivir. No dejes que los ruidos de las opiniones de los demás acallen tu propia voz interior. Y, lo que es más importante, ten el coraje para hacer lo que te dicen tu corazón y tu intuición”

Steve Jobs



## Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al centro Vertex de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Damian Hernández Farramola

---

[Firma autor]

Ing. Jandy Miguel Gómez Rodríguez

---

[Firma tutor]

Ing. Luis Guillermo Silva Rojas

---

[Firma tutor]

## Datos de contacto

Autor:

- Nombre y apellidos: Damian Hernández Farramola
- Correo electrónico: [dfarramola@uci.cu](mailto:dfarramola@uci.cu).
- Institución: Universidad de las Ciencias Informáticas (UCI).
- Centro: Vertex Entornos Interactivos 3D.

Tutores:

- Nombre y apellidos: Luis Guillermo Silva Rojas.
  - Correo electrónico: [lgsilva@uci.cu](mailto:lgsilva@uci.cu).
  - Institución: Universidad de las Ciencias Informáticas (UCI).
  - Centro: Vertex Entornos Interactivos 3D.
- 
- Nombre y apellidos: Jandy Miguel Gómez Rodríguez.
  - Correo electrónico: [jgomez@uci.cu](mailto:jgomez@uci.cu).
  - Institución: Universidad de las Ciencias Informáticas (UCI).
  - Centro: Vertex Entornos Interactivos 3D.

## **Agradecimientos y Dedicatoria**

A mi mamá que siempre ha luchado para encaminarme desde el comienzo de mis estudios y la quiero con la vida, a mi papá que ha formado valores de un verdadero hombre y por eso lo admiro tanto y es mi ídolo. A mi hermana Isnavis que es uno de mis motores impulsores para salir adelante, a mi hermana Verónica que no pudo estar presente en este proceso pero sé que la enorgullezco. A mi abuela Marta por siempre tener fe en que yo alcanzaría mis metas. A mi tía-abuela Zenaida que aunque no se encuentre entre nosotros siempre me entregó su cariño y confianza y por ello le dedico mi tesis, a mi tío-abuelo Figueroa que lo quiero mucho. A toda mi familia que es bastante grande, le agradezco su apoyo y su confianza. A mi novia Dania por siempre estar a mi lado, ayudándome y aguantándome. A mis compañeros de apartamento, que más que compañeros son mis hermanos, a Randy que aunque no me quería de compañero de cuarto por su problema de organización siempre lo querré. A Pi que aunque siempre digo que tengo ganas de salir de él, sé que voy a extrañar sus cambios repentinos de humor que ni él mismo entiende. A mi segunda madre Patricia por ayudarme en todo y formarme como gamer. A mi compañera Sheila que a pesar de su aura tumultuosa la quiero mucho, a mi amiga Leydis por acompañarme en largas horas de videojuegos. A Claudia, a Jaime. A mi gran compañero, profesor integral particular y sobre todo hermano Yasser, por darme toda su ayuda para vencer todos mis obstáculos, a todos mis compañeros de aula que hicieron que esta experiencia fuese única. A mis tutores Guillermo y Jandy por darme todo su apoyo y conocimiento; a mi oponente Yanet, que más que oponente fue una tutora.

## **Resumen**

Con el desarrollo de los ordenadores y el aumento de su potencia y capacidad de cálculo, han surgido ramas como los gráficos por computadora. Dicha rama, al fusionarse con esferas como la industria, la medicina y la arquitectura, brinda, a través de la Realidad Virtual, un apoyo considerable a sus especialistas. Un ejemplo de aplicación de la Realidad Virtual son los paseos virtuales, que permiten al usuario tener una visión tridimensional de un espacio o entorno real. El centro Vertex, Entornos Interactivos 3D, perteneciente a la Universidad de las Ciencias Informáticas, no cuenta con un estándar para la creación de paseos virtuales para la web, que facilite la creación de nuevos paseos, favorezca la inmersión de los usuarios y reduzca su tiempo de despliegue y mantenimiento. En el presente trabajo se hace una propuesta de estándar para las funcionalidades de los paseos virtuales para la web a desarrollar en el centro Vertex. Además, se realiza un estudio del estado del arte, donde se exponen conceptos fundamentales asociados a los paseos virtuales. Como resultado de la investigación, se definieron e implementaron funcionalidades estándar para el desarrollo de paseos virtuales web, que favorecen la interacción e inmersión del usuario y facilitan su mantenimiento. Dentro de estas funcionalidades se destacan la adición de vistas en primera persona, vistas 360 y la carga dinámica de la información de los sitios de interés desde ficheros de texto.

**Palabras clave:** *estándar, paseo, virtual, web.*

## Tabla de contenido

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	4
<b>1.1 Introducción</b> .....	4
<b>1.2 Realidad virtual</b> .....	4
<b>1.3 Paseo virtual</b> .....	5
1.3.1 Paseos virtuales fijos .....	6
1.3.2 Paseos virtuales interactivos .....	7
1.3.3 Paseos virtuales en la web .....	10
<b>1.4 Tecnologías y herramientas utilizadas</b> .....	11
1.4.1 Metodologías de desarrollo de software.....	11
1.4.2 Lenguaje de modelado .....	12
1.4.3 Herramientas utilizadas.....	12
1.4.4 Lenguaje de programación .....	13
1.4.5 Entorno Desarrollo Integrado .....	13
1.4.6 Herramientas de desarrollo .....	14
<b>1.5 Conclusiones parciales</b> .....	18
CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA.....	19
<b>2.1 Propuesta de solución</b> .....	19
<b>2.2 Captura de requisitos</b> .....	21
2.1.2 Requisitos funcionales.....	22
2.1.3 Propuesta de estándar .....	24
2.1.4 Requisitos no funcionales .....	25
<b>2.3 Historias de usuarios</b> .....	26
<b>2.4 Plan de entregas</b> .....	33
2.4.1 Plan de iteraciones.....	34
<b>2.5 Fase de diseño</b> .....	34
2.5.1 Diagrama de clases .....	34
2.5.2 Tarjetas CRC del sistema .....	35
2.5.3 Arquitectura del sistema.....	38
2.5.4 Patrones de diseño .....	40
<b>2.6 Conclusiones parciales</b> .....	42
CAPÍTULO 3: IMPLEMENTACION Y PRUEBAS DEL SISTEMA .....	43
<b>3.1 Implementación</b> .....	43
<b>3.2 Fase de desarrollo</b> .....	43
3.2.1 Estándares de codificación.....	43
3.2.2 Tareas de ingeniería .....	45
3.2.3 Iteración 1 .....	45

3.2.4 Iteración 2.....	48
3.2.5 Iteración 3.....	50
<b>3.3 Fase de pruebas.</b> .....	53
3.3.1 Pruebas de aceptación para las HU de la iteración 1 .....	53
3.3.2 Pruebas de aceptación para las HU de la iteración 2 .....	56
3.3.3 Pruebas de aceptación para las HU de la iteración 3 .....	58
<b>3.4 Conclusiones parciales</b> .....	60
CONCLUSIONES.....	61
RECOMENDACIONES .....	62
REFERENCIAS.....	63
ANEXOS.....	65



## INTRODUCCIÓN

Con el aumento de la potencia de los ordenadores se han desarrollado ramas como los gráficos por computadoras, la cual, al fusionarse con esferas como la industria, la medicina y la arquitectura brindan un apoyo importante a los especialistas de estas ramas a través de la Realidad Virtual (RV)<sup>1</sup>, concepto que cada vez toma más fuerza en la vida cotidiana.

Actualmente, una de las aplicaciones de la RV con auge en el mercado internacional son los Paseos Virtuales<sup>2</sup> (PV). Los Paseos Virtuales más sencillos se basan en imágenes o videos, mientras que otros emplean técnicas más avanzadas como vistas panorámicas, animaciones y modelos tridimensionales de alto nivel de detalle. De forma general, permiten explorar un lugar de forma virtual.

Los PV se utilizan ampliamente en diversas áreas, tales como: Industria del entretenimiento, videojuegos, consolas y turismo. En la Arquitectura se emplean por ejemplo para realizar recorridos internos de edificaciones antes de su construcción o en la representación digital de las instalaciones de una empresa para realizar una visita virtual vía web.

En la Universidad de las Ciencias Informáticas (UCI) se desarrolló un paseo virtual creado por el centro Vertex, Entornos Interactivos 3D. Este paseo se pensó como una aplicación de escritorio (*desktop*) que permitiera un recorrido por algunas áreas de la universidad. Como elementos negativos se puede resaltar la poca portabilidad con la que contaba, el alto nivel de detalle de los modelos 3D generaba mucho espacio de almacenamiento y representaba una sobrecarga para la web, además, todo el paseo fue desarrollado con vistas solo en primera persona, lo que ocasionaba recorridos lentos entre zonas lejanas.

Con el objetivo llegar a un mayor número de usuarios, se elaboró un segundo paseo destinado a la plataforma web, denominado Paseo Virtual UCI Web (PVW), donde, en aras de alcanzar el objetivo trazado, se mitigaron algunas de las deficiencias del PV anterior. Actualmente, el PVW desplegado permite a los usuarios explorar la UCI sin la necesidad de contar con el paseo en su ordenador. La creación del PVW, aunque consiguió su objetivo fundamental de llegar a un mayor número de usuarios, trajo consigo nuevas problemáticas, como su propio mecanismo de interacción, el cual limita la inmersión de los usuarios, al

---

<sup>1</sup> Es un sistema tecnológico que pretende simular las percepciones sensoriales de forma que el usuario las tome como reales.

<sup>2</sup> Es una simulación de un lugar virtual compuesto por una secuencia de imágenes.

contar con solo vistas exteriores. Adicionalmente, la información de los sitios de interés se determina en tiempo de compilación, lo que obliga a un redespigie a la hora de introducir modificaciones.

Teniendo en cuenta la problemática anterior, se plantea como **problema a resolver**: ¿Cómo mejorar la interacción e inmersión del usuario, así como la mantenibilidad de los paseos virtuales para la Web desarrollados en centro Vertex?

Con el objetivo de solucionar el problema planteado con anterioridad, se define como **objeto de estudio** los paseos virtuales. El **campo de acción** se enmarca en el desarrollo de paseos virtuales para plataforma web

Con vistas a resolver el problema identificado, se plantea como **objetivo general**: Definir e implementar las funcionalidades estándar para el desarrollo de paseos virtuales web en el centro Vertex, que mejoren la interacción del usuario y faciliten el mantenimiento.

Para dar cumplimiento al objetivo general, se definieron las siguientes **tareas de investigación**:

1. Elaboración del marco teórico a partir del estado del arte actual relacionado con los paseos virtuales para la web.
2. Análisis de las funcionalidades de diferentes paseos virtuales web para identificar sus principales funcionalidades.
3. Selección de herramientas y lenguajes para la implementación del sistema.
4. Diseño del sistema para guiar el proceso de desarrollo.
5. Implementación del sistema a partir del diseño realizado.
6. Validación del sistema elaborado para verificar si cumple con el objetivo propuesto.

**Este trabajo defiende la siguiente idea**: con la incorporación de nuevas formas de interacción, se mejorará la inmersión del usuario en los PVW del centro Vertex. Adicionalmente, la carga dinámica de la información a mostrar, facilitará las posteriores modificaciones del software.

El resto del documento está estructurado de la siguiente forma:

**Capítulo 1: Fundamentación teórica**: se describe el estado del arte de los paseos virtuales, los principales conceptos relacionados, así como la fundamentación del uso de los lenguajes, herramientas y modelo de desarrollo.

**Capítulo 2: Características y diseño del sistema:** en este capítulo se realiza una descripción de la solución propuesta, se formalizan los requisitos del sistema y se definen las fases de planificación y de diseño. Se elabora además una propuesta de funcionalidades estándar para los paseos virtuales para la web.

**Capítulo 3: Implementación y validación del sistema:** en este capítulo se define la fase de implementación, además, se documentan los diseños de las pruebas realizadas al sistema, así como los resultados obtenidos de su aplicación.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.1 Introducción

En este capítulo se hace alusión a las bases teóricas que sustentan este trabajo de diploma. Para ello, se abordan los conceptos relacionados con el desarrollo de paseos virtuales. Se realiza además un estudio de diferentes PV extranjeros y de los desarrollados en la UCI. Adicionalmente, se describen las tecnologías utilizadas para el desarrollo de la solución propuesta.

### 1.2 Realidad virtual

La RV tuvo sus comienzos en los años 70, como material para una clase de aviación en el departamento de defensa de los Estados Unidos. Inicialmente se empleó para hacer simulaciones de vuelo, con el fin de no arriesgar vidas. Al respecto Ryan (Ryan, 2009) señaló: *“la Realidad Virtual es una experiencia interactiva e inmersiva generada por un ordenador”*.

Otra definición importante sobre el concepto de RV es la del autor Gubern (Gubern, 2009), que manifiesta: *“Un sistema informático que genera entornos sintéticos en tiempo real y que se erigen en la realidad ilusoria, pues se trata de una realidad perceptiva sin soporte objetivo, ya que existe solo dentro del ordenador”*. Ante este pensamiento se interpreta la RV como una representación del mundo real utilizando la tecnología como medio, causando así, la sensación de ser parte de una situación real, donde el usuario pueda interactuar con lo que visualiza.

La RV puede manifestarse de los siguientes tipos (Maldonado, 2008):

**Realidad de tipo inmersiva:** por lo general se encuentran unidas a un ambiente de tres dimensiones creado por computadora, manipulable a través de guantes, cascos electrónicos u otros dispositivos que capturan la posición y los movimientos de distintas partes del cuerpo humano.

**Realidad de tipo no inmersiva:** hace uso de medios como el internet para interactuar en tiempo real con diferentes personas, en espacios y ambientes virtuales, sin necesidad de la integración de dispositivos al computador.

Aunque inicialmente el aprovechamiento de la RV se limitaba exclusivamente a comunidades de investigación académica, militar y de desarrollo; la tecnología de RV ha evolucionado, ampliándose a espacios multidisciplinares, como la educación, el arte, la cultura, el turismo, entre otras. Su uso y aplicación son recomendables para potenciar algún lugar a través de la web. Según Gubern (Gubern, 2009), entre sus beneficios se tiene:

- Supera barreras tanto espaciales como temporales.
- Brinda la posibilidad de la manipulación e interacción.
- Ayuda a un aprendizaje constructivista.
- Constituye una herramienta valiosa para la enseñanza y aprendizaje.
- Proporciona la inmersión en el espacio creado.
- Fomenta el entretenimiento y la distracción.
- Permite un examen detallista de hechos y procesos.
- Permite simular una pauta de actuación.
- Brinda a los usuarios una visualización en 360°.

### **1.3 Paseo virtual**

El paseo virtual es la simulación de un recorrido de un determinado lugar, utilizando tecnología de RV, la expansión de la misma ha sido resultado de largas investigaciones. Actualmente los paseos virtuales se presentan en una multiplicidad de sistemas que permiten al usuario explorar “artificialmente” el lugar en cuestión.

Según Chorafas y cols. (Chorafas, et al., 2009), un paseo virtual es: *“una exploración interactiva de un espacio simulado, con el fin de dar conocimiento de un sitio real sin necesidad de estar presente en éste”*. De acuerdo con la definición brindada en la literatura, un paseo virtual es idóneo para aquellas instituciones o empresas que desean promover un sitio real.

Los paseos virtuales se desarrollan para exponer información a las personas interesadas en los servicios de una institución, de forma tal que muchas empresas, instituciones educativas, entre otras, lo han implementado como forma de publicidad.

El modelado de los paseos virtuales en la web, permite al usuario tener una visión tridimensional de cualquier espacio (calles, edificios, interiores, entre otros). Su visualización abarca desde lugares existentes hasta los no construidos, y su manipulación puede realizarse con el mouse o con las teclas direccionales del computador. Además, este modelado brinda la posibilidad de incorporar herramientas que ejecuten recorridos automáticos, de la misma forma que lo haría un guía real (Ulldemolins, 2011).

### **1.3.1 Paseos virtuales fijos**

Para la realización de un paseo virtual, es preciso conocer cómo presentar las imágenes que se van a mostrar. Es decir, si estas serán fijas o interactivas. Los paseos virtuales fijos impiden que el usuario interactúe con el entorno, por lo que el montaje del paseo virtual determina lo que el usuario observará (Ulldemolins, 2011).

Los tipos de planos que se utilizan al desarrollar un paseo virtual están estrechamente relacionados con el tipo de paseo que se va a implementar. Los paseos virtuales fijos pueden utilizar los planos contenidos en las categorías (Ulldemolins, 2011):

- 1) Planos en movimiento: secuencia en movimiento.
  - 2) Planos 3D: son aquellos que indispensablemente necesitan de la ayuda de un software 3D.
  - 3) Planos clásicos: son planos estáticos heredados del cine. Pueden dividirse en: plano general, plano panorámico general y gran plano general.
- Plano general: introduce al espectador en la situación, le ofrece una vista general y le informa acerca del lugar y de las condiciones en que se desarrolla la acción. Suele colocarse al comienzo de una secuencia narrativa. En un plano general se suelen incluir muchos elementos, por lo que su duración

en pantalla deberá ser mayor que la de un primer plano, para que el espectador pueda orientarse y hacerse cargo de la situación. Puede realizarse de varios modos, según su grado de generalidad.

- Plano panorámico general: es una filmación que abarca muchos elementos muy lejanos. Los personajes tendrán menos importancia que el paisaje; por ejemplo, en una cabaña en el bosque vista de lejos, las personas se verán pequeñas.
- Gran plano general: es una panorámica general con mayor acercamiento de objetos o personas (alrededor de treinta metros).

### **1.3.2 Paseos virtuales interactivos**

Los paseos virtuales interactivos permiten observar diferentes partes de la estructura física de un lugar determinado, e interactuar con los elementos del escenario, realizar alejamientos y acercamientos, caminar por la escena, de tal forma que el usuario maneje el paseo virtual.

Para los paseos virtuales interactivos, se necesita contar con una animación que le permita al usuario interactuar con lo que observa, trasladándose de un lugar a otro, y acercándose o alejándose según lo disponga. Para ello, es necesario conocer qué métodos existen para generar este tipo de recorrido. A continuación, se describen los tipos de planos reportados en la literatura (Ulldemolins, 2011).

#### ➤ **Panorámica 360**

Esta tecnología permite observar el paseo virtual desde una panorámica de 360°, donde el usuario experimenta una sensación parecida a estar en medio de una habitación y girar alrededor de esta, siendo capaz de observar el interior de la habitación.

A este giro se la denomina “panorama 360 cilíndrico”, debido a que lo que realmente hace la tecnología es mapear la imagen del entorno dentro de un cilindro, como si hubiese una cámara en medio de la habitación y se tomase una fotografía continua de todas las paredes, produciendo un efecto de que cada final de una pared sea el principio de la siguiente, lo que produce tener una visualización continua infinita ([Figura 1](#)).



*Figura 1: Mapeado cilíndrico*

Una variante superior de esta tecnología es el “panorama 360 esférico”, el cual hace un mapeo en una esfera en vez de en un cilindro. Esta tecnología es más adecuada cuando se requiere que la vista panorámica recorra por el techo y el piso del escenario, puesto que la continuidad por medio de la navegación es más suave al tratarse de una esfera ([Figura 2](#)).



*Figura 2: Mapeado esférico*



Existen variedad de formatos para realizar tomas cilíndricas o esféricas, el fin siempre será el mismo para ambas tomas, es decir; siempre se proyectará una imagen o conjunto de imágenes donde el final de una parte será el comienzo de otra y de esta forma se le dará la sensación de continuidad.

### ➤ VRML

El lenguaje para modelado de RV (VRML, siglas del inglés *Virtual Reality Modeling Language*) es un formato de archivo normalizado que tiene como objetivo la representación de escenas u objetos interactivos tridimensionales, diseñados particularmente para su empleo en la web.

El lenguaje VRML posibilita la descripción de una escena compuesta por objetos 3D a partir de prototipos basados en formas geométricas básicas o de estructuras en las que se especifican los vértices y las aristas de cada polígono tridimensional y el color de su superficie. Esto permite la navegación por la escena 3D experimentando con los objetos que la componen (Figura 3).

El VRML permite también definir objetos 3D multimedia, a los que se puede asociar un enlace, de manera que el usuario pueda acceder a una página web, imágenes, vídeos u otro fichero VRML de Internet cada vez que haga clic en el componente gráfico en cuestión.



Figura 3: Escena en formato VRML

### 1.3.3 Paseos virtuales en la web

Los paseos virtuales son aplicaciones interactivas muy comunes para plataformas webs, en las que se recrea un elemento o espacio real, generalmente mediante un conjunto de fotografías y modelos tridimensionales. La posibilidad de acceder a estas a través de una página web, permite el acceso universal a los lugares visitados y también permite poner a disposición del visitante información complementaria pero integrada en el paseo.

El desarrollo y evolución de los software de diseño en tres dimensiones posibilita la recreación tridimensional e interactiva de entornos reales, los que se pueden ubicar en cualquier servidor web. El usuario final accede mediante peticiones similares a cualquier página con lenguaje HTML (figura 4).



*Figura 4: Detalle del paseo virtual que recrea, mediante diseño 3D, el Castillo de Santa Catalina (Jaén).*

*Tomado de Gómez Espejo, 2006.*

Sin embargo, en aras de incorporar elementos o espacios más ajustados a la realidad del entorno a mostrar, en el diseño de paseos virtuales orientados a la web, se introducen conceptos y técnicas de fotografía junto al diseño en 3D; además deben cumplir con las siguientes características específicas:

- Balancear el nivel de detalle para reducir el uso de ancho de banda.
- Permitir múltiples usuarios concurrentes.

- Maximizar el número de usuarios potenciales, para esto, las tecnologías empleadas deben ejecutarse en los principales navegadores, sin obligar al usuario a instalar herramientas adicionales.

## 1.4 Tecnologías y herramientas utilizadas

### 1.4.1 Metodologías de desarrollo de software

Para asegurar el éxito durante el desarrollo de software no es suficiente contar con nociones de modelado y herramientas, hace falta un elemento importante: la metodología de desarrollo (Roberth G. Figueroa, 2016). Entre las metodologías de desarrollo de software más usadas en el mundo se encuentran *Rational Unified Process* (RUP), SCRUM y *Extreme Programming* (XP).

**RUP:** se caracteriza por ser una de las más tradicionales, dirigida por casos de uso que definen lo que el usuario desea, a partir de la captura de requisitos y la modelación del negocio. Centrada en la arquitectura, característica que brinda una visión completa del sistema. Además de ser iterativa e incremental, donde el desarrollo se lleva a cabo por fases a través de iteraciones. Aunque la metodología RUP puede adaptarse a cualquier proyecto, no es recomendable aplicarla en proyectos pequeños cuyo objetivo fundamental sea obtener el producto que el cliente necesita en el menor tiempo posible (Roberth G. Figueroa, 2016). Su uso en este tipo de proyectos trae consigo la generación de artefactos que muchas veces no son relevantes.

**SCRUM:** es una metodología de desarrollo que hace énfasis en la gestión de proyecto, permitiendo un control de los procesos que se realizan. No define prácticas de ingeniería de *software* en detalle. Sin embargo, en muchas ocasiones se combina con otras metodologías de desarrollo para suplir sus carencias. Aunque es una metodología ágil, no se beneficia de algunas de las técnicas de desarrollo que este grupo define, por ejemplo: no se especifica la programación en parejas ni las pruebas unitarias (Roberth G. Figueroa, 2016).

**XP:** se encuentra dentro de las metodologías ágiles, planteando la constante retroalimentación con el cliente, teniéndolo como parte del equipo y propiciando la interacción del mismo con el producto en desarrollo. Presenta iteraciones cortas, donde en cada una se obtiene un producto listo para entregar y que tiene valor para el cliente. Es muy flexible a cambios, previendo futuras modificaciones y evitando momentos que paralicen el desarrollo del producto. Además, utiliza prácticas proporcionadas por las metodologías

ágiles, tales como la refactorización, proporcionando un cambio gradual del código fuente hacia un diseño más adaptable. Facilita además la programación en parejas, permitiendo a dos programadores compartir conocimientos técnicos y posibilita una integración continua, asegurando que exista siempre un sistema disponible en el que se ejecuten todas las pruebas (Roberth G. Figueroa, 2016).

Debido a las características de esta investigación, en cuanto al reducido equipo de desarrollo, el corto tiempo para la implementación de la solución y los cambios que pueden producirse a través del proceso de optimización, se considera que la metodología XP es la más factible para guiar este proceso de desarrollo de software. Esta consideración también se debe a que RUP puede desviar el curso de la investigación hacia la generación de artefactos que son irrelevantes.

### **1.4.2 Lenguaje de modelado**

Un modelo se expresa en un determinado lenguaje de modelado. Un lenguaje de modelado gráfico es un lenguaje artificial que sirve para expresar un modelo con símbolos gráficos, en forma de diagramas. Es decir, el modelo “dice” algo acerca del sistema en estudio, y de modo conforme a las reglas del lenguaje de modelado. Esto lleva a comprender que una definición rigurosa del lenguaje de modelado es esencial para obtener todo el fruto posible de los modelos en el proceso de ingeniería (Génova, 2008).

El lenguaje de modelado utilizado es UML 2.0 (por sus siglas en inglés, *Unified Modeling Language*). UML es un lenguaje gráfico que sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas de software como para la arquitectura de hardware donde se ejecuten.

Otro objetivo de este modelado visual es que sea independiente del lenguaje de implementación, de tal forma que los diseños realizados usando UML se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos).

### **1.4.3 Herramientas utilizadas**

Visual Paradigm for UML 8.0: Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML, ideal para ingenieros de software, analistas de sistemas y arquitectos de sistemas

que estén interesados en construcción de sistemas a gran escala y necesiten confiabilidad y estabilidad en el desarrollo orientado a objetos. Además, identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además, permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico (Meza, 2011).

#### **1.4.4 Lenguaje de programación**

Los lenguajes de programación son notaciones para describir la interacción entre las personas y las máquinas. Constituyen un elemento esencial en la construcción de software. Los lenguajes de programación son un método conveniente de describir las estructuras de información y las secuencias de acciones necesarias para ejecutar una tarea concreta. Para el desarrollo de esta aplicación se usará el lenguaje orientado a objetos C# en su versión 6.0. Se escogió un lenguaje orientado a objetos por la facilidad de uso que estos permiten.

##### ➤ Lenguaje de programación C#

C# es un lenguaje con seguridad de tipos y orientado a objetos, que permite a los desarrolladores crear aplicaciones seguras y sólidas que se ejecutan en el *Framework* .NET. La sintaxis de C# simplifica muchas de las complejidades de C++ y proporciona características eficaces, como tipos de valor que aceptan valores NULL, enumeraciones, delegados, expresiones lambda y acceso directo a memoria, que no se encuentran en Java. C# admite métodos y tipo genéricos, que proporcionan una mayor seguridad de tipos y rendimiento, e iteradores, que permiten a los implementadores de clases de colecciones, definir comportamientos de iteración personalizados.

#### **1.4.5 Entorno Desarrollo Integrado**

Un Entorno de Desarrollo Integrado (IDE), es un entorno de programación empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación, tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, entre otros. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto (Sanchez, 2014). Para el desarrollo de la propuesta de solución se usará el IDE MonoDevelop en su versión 6.1, por ser gratuito, ligero y además reconoce las bibliotecas de UnityEngine.

MonoDevelop es un IDE libre y gratuito, diseñado inicialmente para C#. El IDE incluye manejo de clases, ayuda incorporada, completamiento de código, *Stetic* (diseñador de GUI) integrado, soporte para proyectos y un depurador integrado.

### 1.4.6 Herramientas de desarrollo

Las herramientas de diseño gráfico permiten realizar la modelación y manipulación de objetos y entornos virtuales, brindan un conjunto de funciones que facilitan el trabajo con escenarios virtuales y algunas posibilitan la creación de paseos virtuales. El conocimiento de estas herramientas para la investigación es muy importante, ya que permitirá identificar elementos que pudieran ser útiles para realizar la propuesta de solución.

**Blender 2.76:** es la principal suite en código abierto para la creación de contenido 3D, disponible para los principales sistemas operativos (Windows, Linux, Mac) bajo la General Public License (GPL) (Ton Rosendaal, 2009). Entre las ventajas del Blender como programa para el modelado 3D están:

- Posee numerosas herramientas accesibles mediante atajos de teclado, disminuyendo el tiempo de realización.
- Posee la mayoría de las herramientas existentes para modelar en 3D. Ejemplo de esto son las herramientas: Subdividir, Extruir, Rotar, Escalar, Mover, Suavizar, Girar.
- Cuenta con una importante serie de modificadores para lograr efectos deseados en los modelos.

- Permite trabajar con modelos a bajo polígonos, optimizándolos así, para su posterior utilización en la creación de aplicaciones gráficas en tiempo real, como paseos virtuales y video juegos. Blender trabaja con polígonos de 3 o 4 vértices, es decir con triángulos y cuadriláteros (López, 2011).

**Unity 5.0.1:** Unity 3D es un motor gráfico desarrollado por Unity Technologies desde 2001, con el objetivo de permitir a todo el mundo crear atractivos entornos 3D. En los últimos años ha ganado varios premios, entre ellos el “*Wall Street Journal Technology Innovation Award*”, y se encuentra entre las 5 mejores compañías de juegos de 2009 (Baeza, 2015). Unity ofrece la posibilidad de crear escenarios abiertos totalmente editables y manipulables.

Entre las ventajas de usar Unity 3D para el desarrollo de juegos para Android y en general en cualquier proyecto, se pueden mencionar las siguientes:

- Independiente de plataforma: editor disponible para Windows, Linux y Mac. Además permite exportar para las plataformas mencionadas, iOS, Android, WebGL y diferentes consolas.
- Ofrece un editor gráfico de fácil uso y muy potente, el cual permite importar objetos y modelos 3D de forma intuitiva.
- La versión gratuita de la herramienta es muy completa y puede satisfacer las necesidades e intereses para numerosos campos de aplicación.
- Permite la incorporación de nuevas características mediante *plugins* específicos.
- Cuenta con un *Asset Store*, donde se encuentran recursos gratuitos y de pago. El *Asset Store* se accede de manera directa desde el editor de Unity o desde la web.
- Amplia documentación en los foros de la comunidad.

**Unity WebGL:** esta nueva tecnología se ejecuta en el elemento *Canvas* de HTML5, además tiene una integración completa con todas las interfaces DOM (*Document Object Model*). Al ser una API DOM<sup>3</sup>, permite utilizar cualquier lenguaje de programación compatible con DOM como javascript o java.

Según Andreas Gal – CTO de Mozilla: *“Unity siempre ha sido un firme defensor de los juegos web, con la capacidad de hacer la exportación WebGL-plugin gratuito con Unity 5, Mozilla se emociona al ver a Unity promocionarse en la Web como plataforma de primera clase para sus desarrolladores. La exportación One-click to WebGL dará a los desarrolladores de Unity la capacidad de compartir su contenido con una nueva clase de usuarios “.*

Las ventajas de WebGL son las siguientes:

- WebGL se basa en un conocido y ampliamente aceptado estándar de gráficos 3D (OpenGL).
- Compatibilidad con distintos navegadores y plataformas.
- Al ser una API DOM, es compatible con todos los elementos del estándar HTML.
- Gráficos 3D acelerados por hardware (GPU).
- Al utilizar lenguajes de programación interpretados no es necesario compilar el código antes de representarlo y mostrarlo por pantalla.

### **Compatibilidad del Unity WebGL con los navegadores.**

Unity WebGL soporta la mayoría de los navegadores de escritorio en cierto grado. Sin embargo, el nivel de soporte y el rendimiento esperado varía entre los diferentes navegadores. En la siguiente tabla se muestran las características que soportan los principales navegadores.

El contenido de Unity WebGL actualmente no es soportado en los navegadores de dispositivos móviles. Puede que funcione, especialmente en dispositivos de gama alta, pero la mayoría no tienen la memoria

---

<sup>3</sup> Una DOM API es una interfaz de programación para aplicaciones relacionadas con el Document Object Model para acceder y actualizar contenido, estructura y estilo.



necesaria para soportar el contenido de Unity WebGL. Por esta razón, Unity WebGL va a mostrar un mensaje de advertencia cuando intente de cargar contenido en los navegadores móviles (que puede ser desactivado si se necesita).

Esta tabla de compatibilidad es válida para las versiones específicas de los navegadores mencionados. Los navegadores evolucionan rápidamente, por lo que la tabla puede cambiar en las versiones futuras (Unity, 2017).

<b>Matriz de compatibilidad del navegador en desktop</b>					
	<b>Mozilla Firefox 42</b>	<b>Google Chrome 46</b>	<b>Apple Safari 9.0</b>	<b>MS Internet Explorer 11</b>	<b>MS Edge 13</b>
<b>Soporte de WebGL</b> habilidad de correr el contenido de Unity WebGL	X	X	X (Safari 8 y posterior)	X (IE 11 y posterior)	X
<b>Web Audio</b> El Web Audio API es requerido para reproducir sonido en contenido de Unity WebGL.	X	X	X	-	X
<b>Soporte pantalla completa</b>	X	X	-	X	X
<b>Soporte de bloqueo del cursor</b>	X	X	-	-	-
<b>Soporte de Gamepad</b>	X	X	-	-	X
<b>IndexedDB</b> (BD indexada) Requerida para un almacenamiento local como se es utilizada por la característica de Data Caching, la clase PlayerPrefs, y WWW.LoadFromCacheOrDownload.	X	X	X	X	X
<b>WebSockets</b> Requeridos para Networking.	X	X	X	X	X
<b>WebRTC</b> Requerido por la clase WebCamTexture	X	X	-	-	X

Por estas razones, para el desarrollo de la investigación, se decidió emplear Unity WebGL para la exportación del paseo virtual para la plataforma web.

## **1.5 Conclusiones parciales**

Luego de concluido el capítulo, se arriba a las siguientes conclusiones:

- Debido a las características de la investigación, entre las que se encuentra el reducido equipo de desarrollo y el corto tiempo para la implementación de la solución; se decidió utilizar la metodología XP como guía para el desarrollo de la solución.
- Se definió el marco teórico de la investigación a partir del análisis de los conceptos sobre la RV, para lo cual se realizó un estudio referente a los PV y las tecnologías para su desarrollo luego de analizar la bibliografía referente al tema.
- Para el desarrollo de la solución se seleccionaron las siguientes herramientas: Blender para el diseño y modelado 3D, Unity como motor gráfico e IDE (de conjunto con MonoDevelop) y Photoshop para el tratamiento de imágenes.
- Teniendo en cuenta la compatibilidad de WebGL con los principales navegadores web, se decidió el empleo de esta API para el desarrollo de la solución.

## CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DEL SISTEMA

En este capítulo se describe detalladamente el sistema, teniendo en cuenta los requisitos funcionales y no funcionales por los que estará regida la aplicación para su desarrollo, las historias de usuario del sistema y los pasos para darle cumplimiento. Además, se describe la fase de planificación, donde se establecen las prioridades (para el cliente) de cada historia de usuario y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. En esta fase, se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

### 2.1 Propuesta de solución

Como propuesta de solución técnica de este trabajo, se propone el desarrollo de nuevas funcionalidades para el paseo virtual existente para plataforma web. Persiguiendo los objetivos expuestos en el capítulo anterior, se decide integrar la vista en primera persona y vistas panorámicas de locales e importar información de lugares de interés desde un archivo de texto externo, que permita posteriores modificaciones. De esta forma la solución debe permitir:

- Mostrar un menú de opciones de las vistas disponibles para el lugar de interés seleccionado, ver Figura 5.



Figura 5: Menú de opciones.

- Mostrar información del lugar de interés seleccionado. Para ello se propone implementar una función que permita consumir recursos de un servidor. Luego, ubicar en la raíz del paseo, un fichero de texto por cada sitio de interés; seguido, obtener la información del sitio de interés desde el fichero correspondiente y mostrarla en pantalla. De esta manera se pueden realizar modificaciones en la información luego de desplegado el paseo virtual, por lo que supone una mejora para el mantenimiento del mismo, en la Figura 6 se observan los resultados.



Figura 6: Mostrar información.

- Navegar en 1ra persona por el interior de la edificación de interés. Para ello se propone modelar y texturizar el lugar de interés, empleando la herramienta de modelado 3D seleccionada en el epígrafe [1.4.3 Herramientas utilizadas](#). Luego, importar el modelo a Unity, asignar el colisionador a todo el modelo e implementar los controladores de primera persona, para que el usuario pueda navegar por el interior del sitio seleccionado, el prototipo se muestra en la siguiente figura:

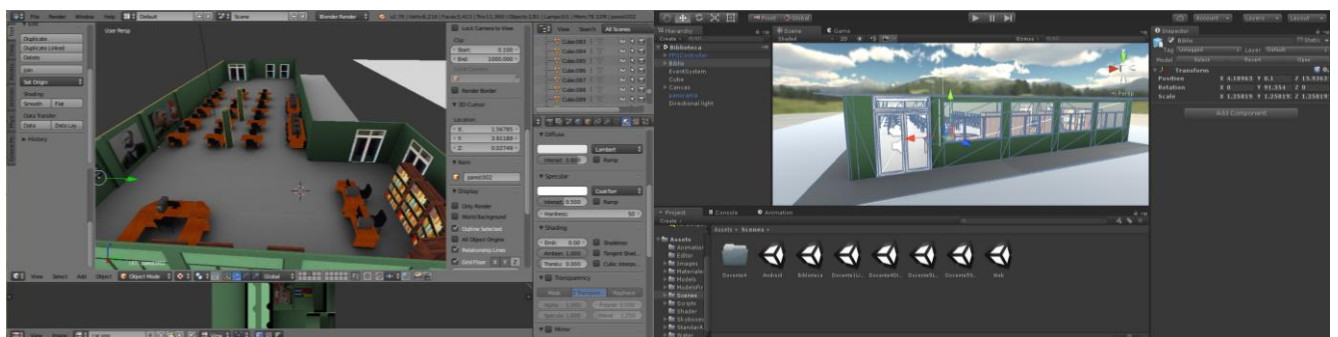


Figura 7: Implementación de vista en 1ra persona.

- Mostrar vista panorámica referente a los locales de interés de la edificación seleccionada. Para ello, se propone crear una esfera con sus normales invertidas y adicionar una cámara de tipo primera

persona en el centro de la esfera. Luego, mapear la imagen panorámica y asignarla como textura de la esfera, la Figura 8 muestra un ejemplo a partir de una imagen panorámica de la librería de la UCI.



*Figura 8: Implementación vista panorámica 360 esférica.*

Esta solución permitirá mayor inmersión e interacción de los usuarios y favorecerá la sensación de presencia dentro del paseo virtual. El objetivo no es sustituir las vistas exteriores, sino contribuir a complementar las opciones de exploración de los sitios de interés.

## **2.2 Captura de requisitos**

La ingeniería de requerimientos, proporciona un mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requerimientos a medida que se transforman en un sistema funcional (Pressman, 2010).

Para la captura de requisitos se emplearon las técnicas: entrevista y estudio bibliográfico o documental. La entrevista permitió adquirir el conocimiento necesario para un mejor dominio del problema, entendimiento del negocio y para definir los objetivos de la solución. El uso de las consultas bibliográficas de varias fuentes de información relacionadas con el objeto de estudio, posibilitó definir nuevas funcionalidades y comprender las diferentes variantes de algoritmos empleados en otras propuestas.

## 2.1.2 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema. Definen la manera que este debe reaccionar a entradas particulares y su comportamiento en situaciones específicas. En algunos casos, los requisitos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. Los requisitos dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar los requerimientos.

A continuación, se describen los requisitos funcionales, definidos para el desarrollo del PVW actualmente en funcionamiento.

Nº	Nombre	Descripción
RF1	Mostrar vista panorámica.	El sistema debe permitir al usuario visualizar todo el campus universitario.
RF2	Visualizar paseo de las esculturas.	El sistema debe permitir al usuario visualizar el paseo de las esculturas.
RF3	Visualizar paseo de los docentes.	El sistema debe permitir al usuario visualizar el paseo de los docentes.
RF4	Listar lugares de interés.	El sistema debe brindar una lista de los lugares de interés de la universidad.
RF5	Seleccionar destino de interés.	El sistema debe permitir al usuario seleccionar el lugar de interés del que desea conocer.
RF6	Visitar lugar de interés.	El sistema debe permitir al usuario visualizar el lugar de interés desde una perspectiva más cerca.
RF7	Mostrar vista Frontal del lugar de interés.	El sistema debe permitir al usuario visualizar el lugar de interés desde su lado frontal.
RF8	Mostrar vista Derecha del lugar de interés.	El sistema debe permitir al usuario visualizar el lugar de interés desde su lado derecho.

RF9	Mostrar vista Trasera del lugar de interés.	El sistema debe permitir al usuario visualizar el lugar de interés desde su parte trasera.
RF10	Mostrar vista Izquierda del lugar de interés.	El sistema debe permitir al usuario visualizar el lugar de interés desde su lado izquierdo.
RF11	Mostrar menú lateral.	El sistema debe mostrarle al usuario un menú lateral con las diferentes acciones que puede realizar.

Teniendo en cuenta los requisitos funcionales anteriores y las deficiencias del PVW, se decide adicionar las siguientes funcionalidades, para ofrecer otros tipos de vistas que favorezcan la interacción e inmersión del usuario con el paseo virtual.

Nº	Nombre	Descripción
RF1	Mostrar menú de opciones de vistas.	El sistema debe permitir al usuario visualizar un menú con las opciones de vistas referentes al lugar de interés seleccionado.
RF2	Mostrar información de sus edificaciones.	El sistema debe permitir al usuario visualizar la información referente al lugar de interés seleccionado.
RF3	Navegar en 1ra persona.	El sistema debe permitir al usuario navegar en 1ra persona por el interior de la edificación de interés.
RF4	Listar vistas panorámicas de los lugares de interés.	El sistema debe brindar una lista de las panorámicas disponibles del sitio de interés seleccionado.
RF5	Navegar en vista panorámica.	El sistema debe permitir al usuario navegar por la vista panorámica referente a los locales de interés de la edificación seleccionada.
RF6	Cargar la información.	El sistema debe cargar la información de los lugares de interés desde un archivo de texto externo.

### 2.1.3 Propuesta de estándar.

Para la confección de la propuesta de estándar para el desarrollo de paseos virtuales para la web en la UCI, se decidió crear una base de requisitos funcionales compuesta por algunos requerimientos del PVW desplegado y las funcionalidades propuestas en esta investigación. Arrojando el siguiente conjunto de funcionalidades, a considerar como estándar para próximos desarrollos de PV para la web:

N°	Nombre	Descripción
RF1	Mostrar vista panorámica.	El sistema debe permitir al usuario visualizar todo el campus universitario.
RF4	Listar lugares de interés.	El sistema debe brindar una lista de los lugares de interés de la universidad.
RF5	Seleccionar destino de interés.	El sistema debe permitir al usuario seleccionar el lugar de interés del que desea conocer.
RF6	Visitar lugar de interés.	El sistema debe permitir al usuario visualizar el lugar de interés desde una perspectiva más cerca.
RF7	Mostrar vista Frontal del lugar de interés.	El sistema debe permitir al usuario visualizar el lugar de interés desde su lado frontal.
RF8	Mostrar vista Derecha del lugar de interés.	El sistema debe permitir al usuario visualizar el lugar de interés desde su lado derecho.
RF9	Mostrar vista Trasera del lugar de interés.	El sistema debe permitir al usuario visualizar el lugar de interés desde su parte trasera.
RF10	Mostrar vista Izquierda del lugar de interés.	El sistema debe permitir al usuario visualizar el lugar de interés desde su lado izquierdo.



RF11	Mostrar menú lateral.	El sistema debe mostrarle al usuario un menú lateral con las diferentes acciones que puede realizar.
RF12	Mostrar menú de opciones de vistas.	El sistema debe permitir al usuario visualizar un menú con las opciones de vistas referentes al lugar de interés seleccionado.
RF13	Mostrar información de los lugares de interés.	El sistema debe permitir al usuario visualizar la información referente al lugar de interés seleccionado.
RF14	Navegar en 1ra persona.	El sistema debe permitir al usuario navegar en 1ra persona por el interior de la edificación de interés.
RF15	Listar vistas panorámicas de los lugares de interés.	El sistema debe brindar una lista de las panorámicas disponibles del sitio de interés seleccionado
RF16	Navegar en vista panorámica.	El sistema debe permitir al usuario navegar por la vista panorámica referente a los locales de interés de la edificación seleccionada.
RF17	Cargar la información.	El sistema debe cargar la información de los lugares de interés desde un archivo de texto externo.

#### **2.1.4 Requisitos no funcionales**

Los requisitos no funcionales, como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema, como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema. A continuación, se detallan los requisitos no funcionales definidos para este trabajo:

- **Requerimientos de Software:** el sistema debe permitir su ejecución en los navegadores Mozilla Firefox 42, Google Chrome 46, Internet Explorer 11, Safari 9.0, o versiones superiores de estos navegadores.
- **Requerimientos de Hardware:** se recomienda como mínimo una PC con CPU Intel Pentium IV/AMD Athlon 64 3000+, 1 Gb de RAM.
- **Restricciones en el diseño y la implementación:** lenguaje de programación C#.
- **Requerimientos de usabilidad:** permitir una navegación sencilla, tanto para usuarios con conocimientos avanzados de informática, como para inexpertos. Esto se logrará a partir de una arquitectura de información simple, donde el usuario, en todo momento, tendrá conocimientos del lugar donde se encuentra dentro del sistema.

## 2.3 Historias de usuarios

Las historias de usuario (HU) tienen gran similitud con los casos de uso en RUP, con la diferencia de que deben ser descritas en no más de tres líneas e idealmente es el cliente quien las redacta y prioriza. Posteriormente se le suma un tiempo estimado de desarrollo, que lo define el propio equipo de desarrollo. En esencia, no son más que las ideas del cliente, organizadas y agrupadas de acuerdo a su funcionalidad. También se tiene en cuenta un orden tal que permita al cliente priorizar sus necesidades y al equipo de trabajo definir las que resultan críticas o claves en el desarrollo de la solución.

Las HU también conducen el proceso de creación de los *tests* de aceptación (empleados para verificar que las HU se implementaron correctamente). Existen diferencias entre las HU y la tradicional especificación de requisitos, fundamentalmente en el nivel de detalle. Las HU solamente proporcionaran los detalles sobre la estimación del riesgo y del tiempo empleado para su implementación.

En la claridad de su descripción radica el éxito del proyecto. La falta de comunicación y la comprensión errónea de las necesidades del cliente son la principal causa de fracaso en proyectos pequeños. Es por ello que las HU juegan un papel importante en este proceso y se les dedica un esfuerzo considerable para su realización efectiva. A continuación, se muestran las HU definidas para la solución propuesta:

Tabla 1: HU Nro.1 Mostrar menú de opciones.

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre:</b> Mostrar menú de opciones.
<b>Usuario:</b> Usuario General	
<b>Prioridad en el negocio:</b> Alta	<b>Iteración asignada:</b> 1
<b>Tiempo estimado:</b> 1 semana	
<b>Programador responsable:</b> Damian Hernández Farramola	
<p><b>Descripción:</b>                      El usuario selecciona el lugar de interés deseado.                      El sistema muestra una vista con el menú de opciones: “Vista en 1ra persona”, “Vistas panorámicas”, “Liberar cámara”, “Visitar” e “Información”.</p>	
<b>Observaciones:</b> No todos los lugares de interés tienen las opciones de los tipos de vistas habilitadas.	
<b>Prototipo de Interfaz:</b>	
 <p>The image shows a 3D perspective view of a city with various buildings, roads, and green spaces. In the top right corner, there are two circular icons: one with a person silhouette and another with a camera viewfinder. On the right side, there is a vertical stack of four circular icons: a camera, an information 'i' symbol, and a classical building facade.</p>	

Tabla 2: HU Nro.2 Mostrar información de los lugares de interés.

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre:</b> Mostrar información de los lugares de interés
<b>Usuario:</b> Usuario General	
<b>Prioridad en el negocio:</b> Alta	<b>Iteración asignada:</b> 1
<b>Tiempo estimado:</b> 2 semanas	
<b>Programador responsable:</b> Damian Hernández Farramola	
<p><b>Descripción:</b>                      El usuario selecciona el lugar de interés deseado.                      El sistema muestra una vista con el menú de opciones, entre las que se encuentra la opción “Información”.                      El usuario selecciona la opción “Información”.                      El sistema muestra la información referente al lugar de interés seleccionado en un panel ubicado en el inferior de la interfaz. Si el usuario desea dejar de visualizar dicha información, debe presionar la opción “Información” nuevamente.</p>	
<b>Observaciones:</b> N/A	
<p><b>Prototipo de Interfaz:</b></p> 	

Tabla 3: HU Nro.3 Navegar en 1ra persona por el interior de la edificación de interés

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre:</b> Navegar en 1ra persona por el interior de la edificación de interés.
<b>Usuario:</b> Usuario General	
<b>Prioridad en el negocio:</b> Alta	<b>Iteración asignada:</b> 3
<b>Tiempo estimado:</b> 3 semanas	
<b>Programador responsable:</b> Damian Hernández Farramola	
<p><b>Descripción:</b>                      El usuario selecciona el lugar de interés deseado.                      El sistema muestra una vista con el menú de opciones: “Vista en 1ra persona”, “Vistas panorámicas”, “Liberar cámara”, “Visitar” e “Información”.                      El usuario selecciona la opción “Vista en 1ra persona”.                      El sistema muestra la escena en 1ra persona del lugar de interés seleccionado.                      El usuario puede navegar por la escena en 1ra persona haciendo uso de las teclas A, S, D, W, ←, ↑, →, ↓ o mediante los botones que brinda la interfaz. Si el usuario desea regresar a la interfaz principal, debe seleccionar el botón regresar ubicado en la esquina inferior izquierda de la interfaz.</p>	
<b>Observaciones:</b> No todos los lugares de interés tienen esta opción de vista en 1ra persona	

**Prototipo de Interfaz:**

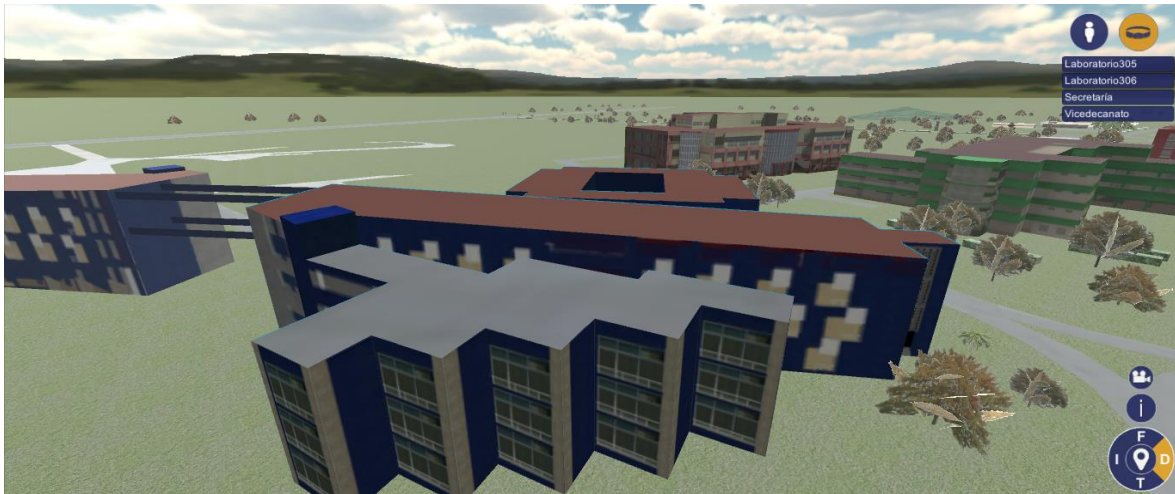


*Tabla 4: HU Nro.4 Listar vistas panorámicas del lugar de interés.*

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre:</b> Listar vistas panorámicas del lugar de interés.
<b>Usuario:</b> Usuario General	
<b>Prioridad en el negocio:</b> Alta	<b>Iteración asignada:</b> 1
<b>Tiempo estimado:</b> 1 semana	
<b>Programador responsable:</b> Damian Hernández Farramola	
<p><b>Descripción:</b>                      El usuario selecciona el lugar de interés deseado.                      El sistema muestra una vista con el menú de opciones: “Vista en 1ra persona”, “Vistas panorámicas”, “Liberar cámara”, “Visitar” e “Información”.                      El usuario selecciona la opción “Vistas panorámicas”.                      El sistema despliega un menú con las opciones de vistas panorámicas correspondientes al lugar de interés. Si el usuario desea dejar de visualizar el listado de las panorámicas puede presionar nuevamente la opción “Vistas panorámicas” o la opción liberar cámara.</p>	

**Observaciones:** No todos los lugares de interés tienen esta opción habilitada.

**Prototipo de Interfaz:**



*Tabla 5: HU Nro.5 Navegar en la vista panorámica seleccionada.*

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre:</b> Navegar en la vista panorámica seleccionada.
<b>Usuario:</b> Usuario General	
<b>Prioridad en el negocio:</b> Alta	<b>Iteración asignada:</b> 3
<b>Tiempo estimado:</b> 2 semanas	
<b>Programador responsable:</b> Damian Hernández Farramola	



**Descripción:**

El usuario selecciona el lugar de interés deseado.

El sistema muestra una vista con el menú de opciones: "Vista en 1ra persona", "Vistas panorámicas", "Liberar cámara", "Visitar" e "Información".

El usuario selecciona la opción "Vistas panorámicas".

El sistema despliega un menú con las opciones de vistas panorámicas correspondientes al lugar de interés.

El usuario selecciona una de las vistas panorámicas.

El sistema muestra la escena de la vista panorámica seleccionada. El usuario puede navegar en la panorámica mediante el uso del mouse. Si el usuario desea regresar a la interfaz principal, debe presionar el botón "Regresar" ubicado en la esquina inferior izquierda de la interfaz.

**Observaciones:** No todos los lugares de interés tienen las opciones de los tipos de vistas habilitadas.

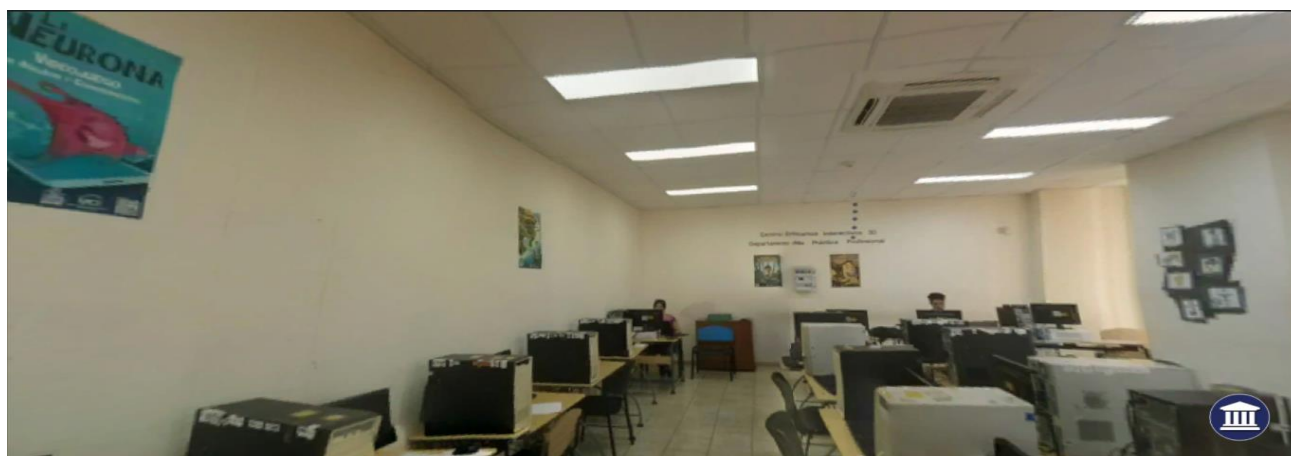
**Prototipo de Interfaz:**

Tabla 6: HU Nro.6 Cargar la información de los lugares de interés desde un archivo de texto configurable.

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre:</b> Cargar la información de los lugares de interés desde un archivo externo con extensión .txt.
<b>Usuario:</b> Usuario General	



<b>Prioridad en el negocio:</b> Alta	<b>Iteración asignada:</b> 2
<b>Tiempo estimado:</b> 2 semanas	
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> El sistema debe de ser capaz de cargar la información que muestra referente a los lugares de interés desde un archivo externo con extensión <b>.txt</b> .	
<b>Observaciones:</b> N/A	
<b>Prototipo de Interfaz:</b> N/A	

## 2.4 Plan de entregas

El plan de entregas tiene el objetivo fijar el período de tiempo que se puede tardar en la implementación de cada una de las HU. En la primera iteración se intentará establecer una arquitectura del sistema que pueda utilizarse durante el resto del proyecto. Esto se logra escogiendo las HU que fueren la creación de esta arquitectura, las HU escogidas se observan en la Tabla 8.

*Tabla 4: Plan de Entregas*

Entregas	Historias de Usuarios	Semanas
<b>Entrega 1</b>	Mostrar menú de opciones.	1
	Listar vistas panorámicas del lugar de interés.	1
	Navegar en 1ra persona por el interior de la edificación de interés.	3
<b>Entrega 2</b>	Cargar la información de los lugares de interés desde un archivo externo con extensión <b>.txt</b> .	2
	Proyectar información referente a los lugares de interés.	2
<b>Entrega 3</b>	Navegar en la vista panorámica seleccionada.	3
<b>Total</b>		12

### 2.4.1 Plan de iteraciones

En la siguiente tabla se muestran las iteraciones definidas para el desarrollo del sistema y las entregas del producto que se realizarán en cada una de estas (Tabla 9).

*Tabla 5: Plan de Iteraciones*

Iteración 1 3ra semana de enero	Iteración 2 3ra semana de marzo	Iteración 3 3ra semana de mayo
<b>Entrega 1</b>	<b>Entrega 2</b>	<b>Entrega 3</b>

## 2.5 Fase de diseño

El diseño del sistema crea una estructura para la organización de la lógica del sistema y posibilita que sea escalable con la menor cantidad de cambios posible. Los diseños, se recomienda que sean simples, en caso de tornarse complejos, se sugiere dividirlos en componentes más pequeños.

### 2.5.1 Diagrama de clases

Los diagramas de clases son diagramas de estructuras estáticas que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, entre otros). Además, son el pilar básico del modelado UML, se utilizan tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). El diagrama de clases de más alto nivel, será lógicamente un dibujo de los paquetes que componen el sistema. Las clases se documentan con una descripción de lo que hacen, sus métodos y sus atributos. Las relaciones entre clases se documentan con una descripción de su propósito, de los objetos que intervienen en sus relaciones y su opcionalidad (Leon, 2013).

En la figura 9 se muestra el diagrama de clases correspondiente al sistema, reflejando los elementos que conforman la solución.

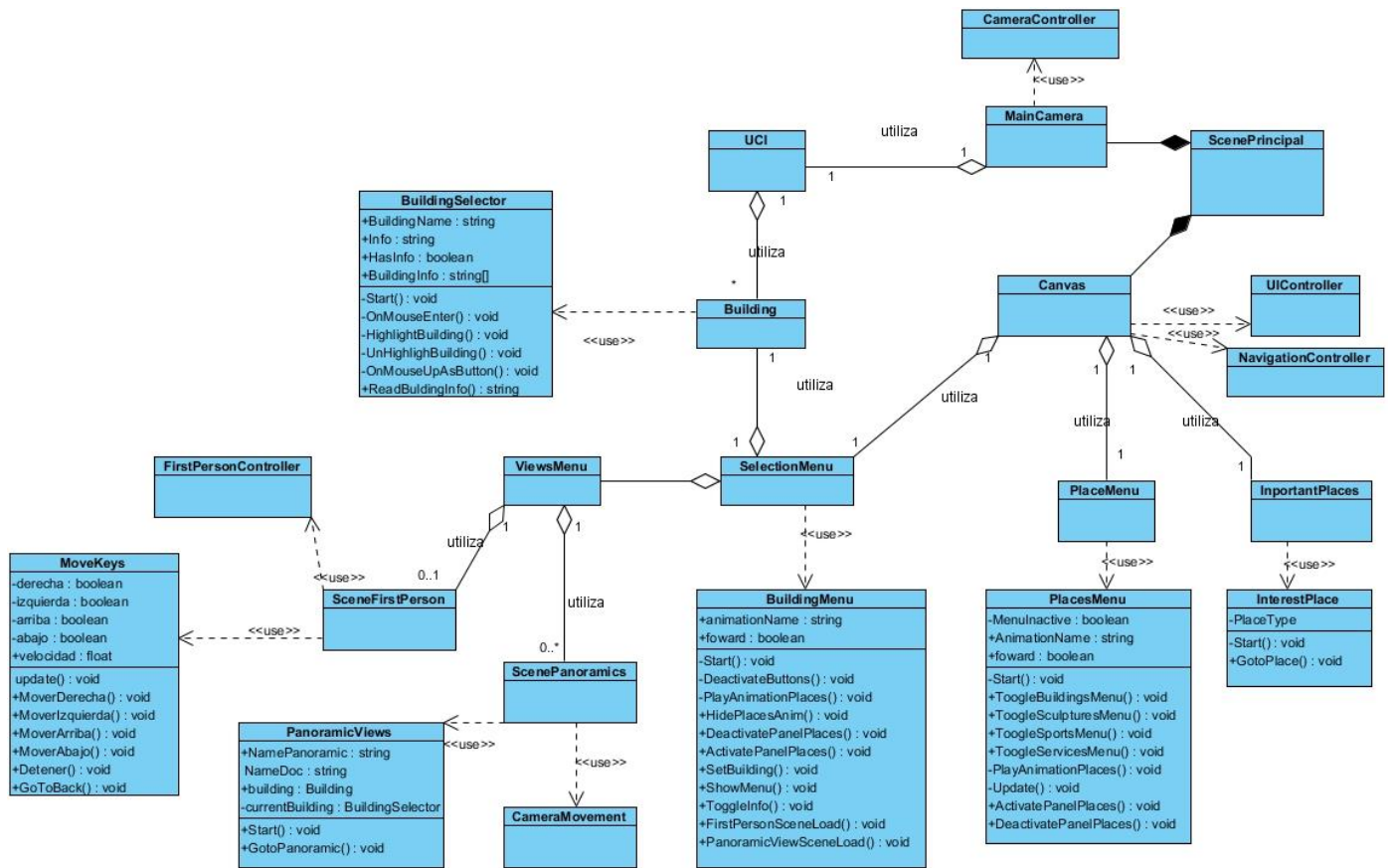


Figura 9: Diagrama de Clases

## 2.5.2 Tarjetas CRC del sistema

La metodología XP en lugar de utilizar diagramas para desarrollar modelos, representa las clases mediante tarjetas. Las tarjetas CRC (Clase, Responsabilidad y Colaboración) ayudan al equipo a definir actividades durante el diseño del sistema. Cada tarjeta representa una clase en la programación orientada a objetos y define sus responsabilidades (lo que ha de hacer) y las colaboraciones con las otras clases (cómo se comunica con ellas). La plantilla para la elaboración de las tarjetas CRC se encuentra especificada en los anexos (Tabla 10), a partir de esta plantilla, se generaron las siguientes tarjetas CRC:

Tabla 6: Tarjeta CRC de la clase "BuildingSelector"

Tarjeta CRC	
<b>Clase:</b> "BuildingSelector.cs"	
Responsabilidad:	Colaboración:
Start()	BuildingMenu
OnMouseEnter()	InterestPlace
HighlightBuilding()	
OnMouseUpButton()	
UnHighlightBuilding()	
ReadBuildingInfo()	

Tabla 7: Tarjeta CRC de la clase "BuildingMenu"

Tarjeta CRC	
<b>Clase:</b> "BuildingMenu.cs"	
Responsabilidad:	Colaboración:
Start()	PlacesMenu
DeactivateButtons()	
PlayAnimationPlaces()	
HidePlacesAnim()	
DeactivatePanelPlaces()	
ActivatePanelPlaces()	

SetBuilding()	
ShowMenu()	
ToggleInfo()	
FirstPersonSceneLoad()	
PanoramicViewSceneLoad()	

*Tabla 8: Tarjeta CRC de la clase "PlacesMenu".*

Tarjeta CRC	
<b>Clase:</b> "PlacesMenu.cs"	
<b>Responsabilidad:</b>	<b>Colaboración:</b>
Start()	InterestPlace
ToogleBuildingsMenu()	
ToogleSculpturesMenu()	
ToogleSportsMenu()	
ToogleServicesMenu()	
PlayAnimationPlaces()	
Update()	
ActivatePanelPlaces()	
DeactivatePanelPlaces()	

Tabla 9: Tarjeta CRC de la clase "InterestPlace"

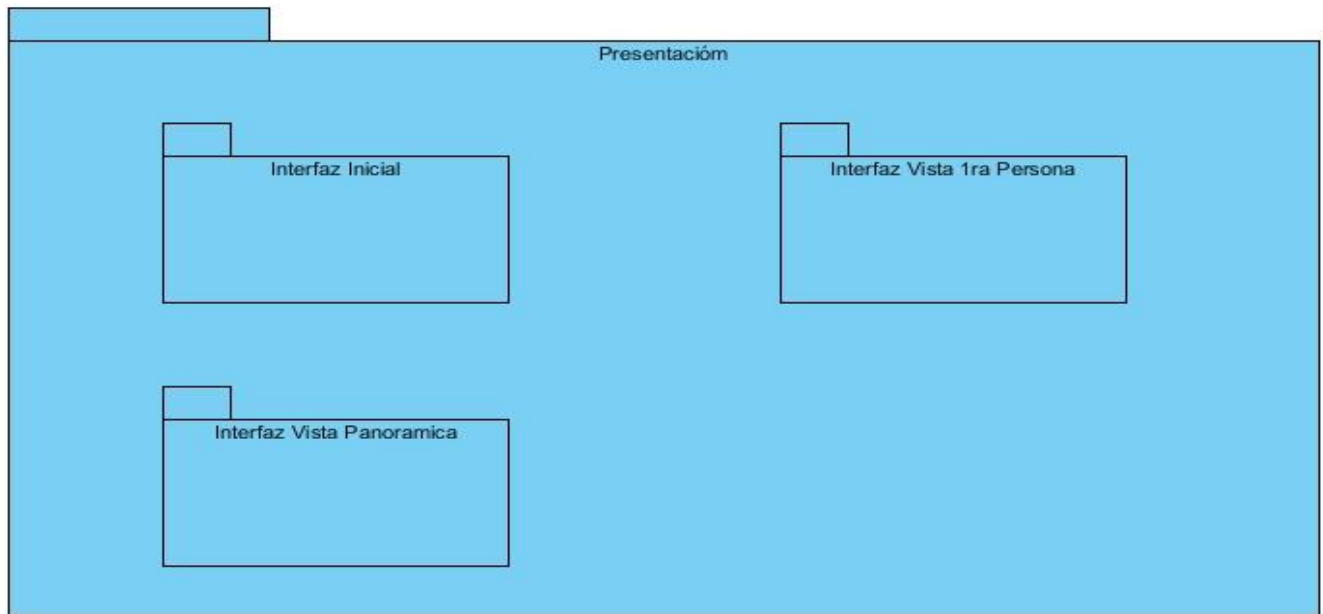
Tarjeta CRC	
Clase: "InterestPlace.cs"	
Responsabilidad:	Colaboración:
Start()	
GotoPlace()	

### 2.5.3 Arquitectura del sistema

Los patrones de arquitectura son aquellos que expresan un esquema organizativo estructural, fundamental para sistemas de software.

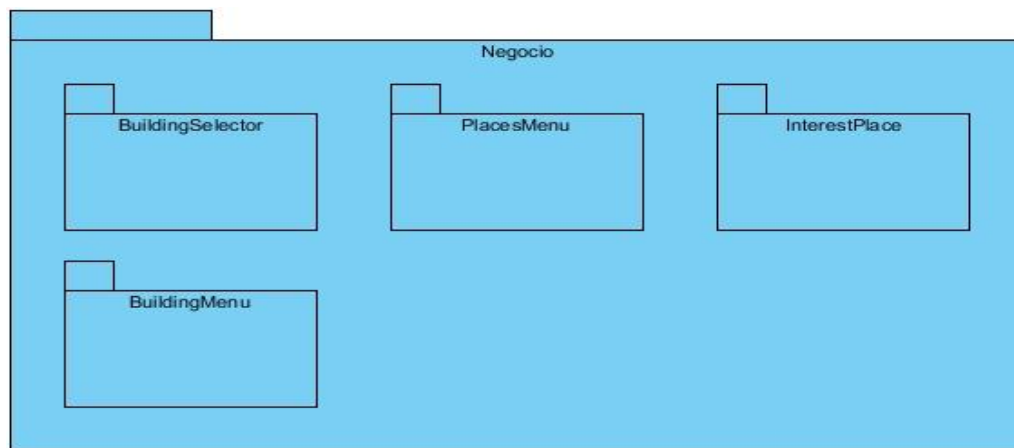
La creación del PVW se desarrolló sobre el motor gráfico *Unity* y responde a una arquitectura basada en capas, compuesta por la capa de presentación, capa de negocio y capa de soporte. Los sistemas o arquitecturas en capas, brindan entre otros beneficios un cierto aislamiento, en caso de ocurrir un error o de que sea necesario actualizar algún elemento, solo se realizan cambios en la capa correspondiente y el resto se mantiene intacto. Por tanto, con el objetivo de continuar con el desarrollo del PVW se decidió utilizar la arquitectura antes mencionada.

**Capa de presentación:** es la capa con la que interactúa el usuario y está formada por las interfaces, las cuales le permiten al usuario navegar por el sistema (Figura 10).



*Figura 10: Capa de presentación*

**Capa de negocio:** está formada por las entidades, que representan objetos que van a ser manejados o utilizados por la aplicación (Figura 11).



*Figura 11: Capa de negocio*

**Capa de soporte:** contiene las bibliotecas para la física, los gráficos, el sonido, y el almacenamiento de los datos brindados por el motor gráfico de *Unity* (Figura 12).

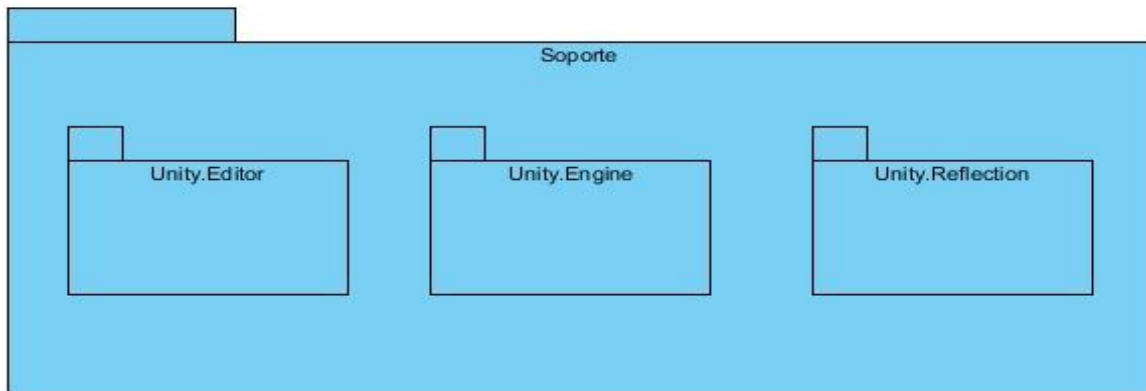


Figura 12: Capa de soporte

#### 2.5.4 Patrones de diseño

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra característica es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (Gamma, 1995).

Los Patrones de Asignación de Responsabilidades (GRASP) da unos principios generales para asignar responsabilidades y se utiliza sobre todo en la realización de diagramas de interacción entre clases (Carmona, 2012).

**Patrón experto:** asigna la responsabilidad al experto en información, que es la clase que cuenta con la información necesaria para cumplir dicha responsabilidad (Figura 10). El empleo adecuado de este patrón proporciona muchos beneficios, uno de ellos es que se conserva el encapsulamiento, debido a que los objetos se valen de su propia información para hacer lo que se pide; esto soporta un bajo acoplamiento y permite tener sistemas más robustos. Además, brinda soporte a una alta cohesión, al distribuir el comportamiento entre las clases que cuentan con la información requerida, facilita que las definiciones de dichas clases sean más sencillas y cohesivas.



En la implementación del PVW este patrón se aplica en la entidad “BuildingSelector” ya que tiene los datos de inicialización que serán transmitidos a otras entidades como a “BuildingMenu”.

```
1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.UI;
4
5 public class BuildingSelector : MonoBehaviour
6
7     [Header("Información del edificio")]
8     public string BuildingName = "PlaceX";
9     public bool HasInfo;
10    public string Info;
11    [TextArea(3, 20)] public string BuildingInfo;
12
13    [Header("Opciones")]
14    public bool HasScene;
15    public string SceneName;
16
17    [Header("Vistas")]
18    public bool HasViews;
19    public Transform NorthView;
20    public Transform EastView;
21    public Transform SouthView;
22    public Transform WestView;
23    [Header("Damian")]
24    public GameObject TextPanelInfo;
25
26
```

Figura 13: Fragmento de código donde se emplea el patrón experto

**Patrón bajo acoplamiento:** El acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma, una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Las diferentes clases controladoras sólo dependen de un único controlador para realizar sus funcionalidades (Mestras, 2004). Este patrón se tuvo presente debido a la importancia que se le atribuye a realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y que a su vez permitan la reutilización.

**Patrón alta cohesión:** Se aplica para realizar un diseño que evite contener clases con un alto grado de abstracción, que asuman responsabilidades que podían haber delegado a otros objetos o que tengan responsabilidades muy complejas. Se tienen las clases controladoras que se encargan de ejecutar acciones

de acuerdo a las peticiones que le llegan, y las clases de acceso a datos que interactúan con el modelo, de forma tal que se elimina la sobrecarga de funcionalidades en las clases controladoras (Mestras, 2004). La clase PlacesMenu contiene un método llamado PlayAnimationPlaces, el cual se llama desde la clase controladora InterestPlace pero se ejecuta en la clase PlacesMenu, librando a la controladora de esta responsabilidad.

**Patrón controlador:** Es un evento generado por actores externos. Se asocian con operaciones del sistema como respuestas a los eventos del sistema, tal como se relacionan los mensajes y los métodos. Normalmente, un controlador delega en otros objetos el trabajo que se necesita hacer; coordina o controla la actividad. No realiza mucho trabajo por sí mismo. En este trabajo existe la clase controladora InterestPlace que es la encargada de ubicar al usuario en el lugar de interés.

## 2.6 Conclusiones parciales

Tras la definición de las características del sistema se arrojan las siguientes conclusiones:

- Se define la propuesta de estándar para desarrollo de paseos virtuales web en la universidad, que permite contar con una guía de funcionalidades a implementar en el desarrollo de futuros PV.
- Se especificaron los requisitos funcionales y no funcionales con que cuenta la propuesta de solución, los que servirán de guía para el desarrollo de la aplicación informática.
- Se realiza la descripción, mediante historias de usuarios, que se implementarán en tres iteraciones. Además, se definieron las tarjetas CRC necesarias para establecer el orden de implementación de las HU.
- Se definió la arquitectura y se describió la propuesta de solución a través de sus principales funcionalidades.

## CAPÍTULO 3: IMPLEMENTACION Y PRUEBAS DEL SISTEMA

### 3.1 Implementación

XP define un desarrollo por iteraciones, realizándose, en cada una, la implementación de las historias de usuario establecidas en la etapa de diseño. Este desarrollo se encuentra guiado por pruebas, que se deben realizar cuando se termina la implementación de cada historia de usuario. Uno de los aspectos más importantes a tener en cuenta para la implementación de las HU es el uso de estándares de codificación, de forma tal que se mantenga el código consistente y facilite su comprensión y escalabilidad.

Para comenzar la implementación del producto, XP propone tener como base una arquitectura lo más flexible posible para evitar grandes cambios en las próximas iteraciones y en las modificaciones que propone el cliente. Teniendo en cuenta que la presente solución está enfocada a los detalles prácticos, de forma que estos puedan ser recopilados como un estándar, es necesario definir correctamente la arquitectura del software (Ver epígrafe 2.5.3). Una vez trazada la misma, se procede a la primera iteración.

### 3.2 Fase de desarrollo

En esta fase, los clientes y los desarrolladores del proyecto deben estar en comunicación, para que los desarrolladores puedan implementar las funcionalidades pactadas. En esta etapa está incluida toda la codificación o programación realizada por los desarrolladores del sistema. Para lograr un correcto equilibrio en el código y facilitar el entendimiento y futuro mantenimiento, se deben seguir un conjunto de estándares de codificación.

#### 3.2.1 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código; si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. Por otra parte, la mantenibilidad es la facilidad con que el

sistema puede modificarse para añadirle nuevas características, modificar las existentes, depurar errores o mejorar el rendimiento (Microsoft, 2015).

Con el objetivo de lograr una estandarización en la implementación, se decide aplicar el estilo de escritura empleado en estándares de codificación, lo cual facilitará su posterior soporte y mantenimiento, así como una mejor lectura y comprensión del código. A continuación, se especifican los estándares de codificación que fueron utilizados en el desarrollo de la propuesta de solución:

#### **Nomenclatura de las clases:**

- El nombre de las clases comenzará con mayúscula, si este está compuesto por varias palabras, las demás comenzarán con mayúscula también. Ejemplo: *“PanoramicViews”*, *“MoveKey”*.
- Se deberán redactar los nombres de las clases descriptivos y simples. Usar palabras completas, evitar acrónimos y abreviaturas, a no ser que la abreviatura sea mucho más conocida que el nombre completo, como URL o HTML.

#### **Indentación:**

- Para la indentación de contenidos se utilizarán *tabs* nunca espacios en blanco.

#### **Nombre de variables:**

- Los nombres de las variables se deberán redactar evitando ambigüedades y se evitarán los nombres que no sean de fácil comprensión o que no tengan ningún sentido.

#### **General:**

- Se evita el uso de las tildes y la letra ñ.
- Se utilizarán nombres que sean claros, concretos y libres de ambigüedades.

### 3.2.2 Tareas de ingeniería

A continuación, se definen las tareas de ingeniería para cada historia de usuario, agrupadas según la iteración a la que pertenecen. Estas tareas tienen como objetivo definir de manera simple cada una de las actividades necesarias para dar cumplimiento a las HU.

### 3.2.3 Iteración 1

El principal objetivo de la primera iteración es desarrollar la **HU Nro. 1**: Proyectar menú de opciones, **HU Nro. 5**: Proyectar menú desplegable de panorámicas y **HU Nro. 3**: Navegar en 1ra persona por el interior de la edificación de interés.

Esta iteración arroja como resultado la versión 1.0 del producto, permitiendo que el usuario interactúe con el menú de opciones de vistas y navegue en 1ra persona por el interior del lugar de interés seleccionado, utilizando para esto las teclas A, S, D, W, ←, ↑, →, ↓ o mediante los botones que brinda la interfaz para el usuario. Para su implementación se trazaron las tareas que se describen a continuación:

1. Diseño de los botones del menú de opciones.
2. Animación de los botones del menú de opciones.
3. Implementación e integración de las funcionalidades de los botones del menú de opciones al paseo virtual.
4. Diseño del menú desplegable.
5. Animación del menú desplegable.
6. Implementación e integración de la funcionalidad del menú desplegable al paseo virtual.
7. Modelado el interior de un lugar de interés, en este caso se escogió la Biblioteca.
8. Diseño de los botones para la navegación en 1ra persona.
9. Implementación del controlador de 1ra persona.
10. Implementación de la funcionalidad de los botones para el controlador de 1ra persona.

Tabla 15: Tarea de ingeniería Nro.1

Tarea de Implementación	
<b>Número:1</b>	<b>Número de HU:1</b>
<b>Nombre de la tarea:</b> Diseño de los botones del menú de opciones	
<b>Tipo de tarea:</b> Diseño	<b>Tiempo estimado:</b> 2 días
<b>Fecha de inicio:</b> 25 de diciembre de 2016	<b>Fecha de fin:</b> 27 de diciembre de 2016
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se diseñan los botones correspondientes a las funciones que se añadirán en el paseo virtual, manteniendo el patrón de diseño de dicho paseo.	

Tabla 16: Tarea de ingeniería Nro.2

Tarea de Implementación	
<b>Número:2</b>	<b>Número de HU:1</b>
<b>Nombre de la tarea:</b> Animación de los botones del menú de opciones.	
<b>Tipo de tarea:</b> Diseño	<b>Tiempo estimado:</b> 1 día
<b>Fecha de inicio:</b> 27 de diciembre de 2016	<b>Fecha de fin:</b> 28 de diciembre de 2016
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se realiza la animación de entrada a la interfaz de los botones del menú de opciones.	

Tabla 17: Tarea de ingeniería Nro.3

Tarea de Implementación	
<b>Número:</b> 3	<b>Número de HU:</b> 1
<b>Nombre de la tarea:</b> Implementación e integración de las funcionalidades de los botones del menú de opciones al paseo virtual.	
<b>Tipo de tarea:</b> Implementación	<b>Tiempo estimado:</b> 2 días
<b>Fecha de inicio:</b> 28 de diciembre de 2016	<b>Fecha de fin:</b> 30 de diciembre de 2016
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se implementa la funcionalidad de los botones que permite al usuario seleccionar un tipo de vista para el lugar de interés previamente elegido.	

Tabla 18: Tarea de ingeniería Nro.4

Tarea de Implementación	
<b>Número:</b> 4	<b>Número de HU:</b> 5
<b>Nombre de la tarea:</b> Diseño del menú desplegable.	
<b>Tipo de tarea:</b> Diseño	<b>Tiempo estimado:</b> 2 días
<b>Fecha de inicio:</b> 30 de diciembre de 2016	<b>Fecha de fin:</b> 1 de enero de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	

**Descripción:**

Se diseña el menú desplegable siguiendo el patrón de diseño del paseo virtual, que permita mostrar en la interfaz las opciones de vistas panorámicas a elegir por el usuario.

*Tabla 19: Tarea de ingeniería Nro.5*

Tarea de Implementación	
<b>Número:</b> 5	<b>Número de HU:</b> 5
<b>Nombre de la tarea:</b> Animación del menú desplegable.	
<b>Tipo de tarea:</b> Diseño	<b>Tiempo estimado:</b> 1 días
<b>Fecha de inicio:</b> 1 de enero de 2016	<b>Fecha de fin:</b> 2 de enero de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se anima la entrada en interfaz del menú desplegable	

Las restantes tareas de ingeniería de la iteración 1, están especificadas en los [anexos](#).

### 3.2.4 Iteración 2

El principal objetivo de la segunda iteración es desarrollar la **HU Nro. 7**: Cargar la información de los lugares de interés desde un archivo externo con extensión .txt y la **HU Nro. 2**: Proyectar información de los lugares de interés. Esta iteración aporta como resultado la versión 1.5 del producto, permitiendo al usuario visualizar la información referente a los lugares de interés previamente seleccionados por el mismo. Para su implementación se trazaron las tareas que se describen a continuación:

1. Creación de un fichero con extensión .txt por cada lugar de interés, este fichero debe contener la información de los sitios de interés.



2. Implementación de la carga dinámica de la información de los ficheros externos.
3. Implementación e integración de la funcionalidad “proyectar información”.

*Tabla 25: Tarea de ingeniería Nro.11*

Tarea de Implementación	
<b>Número:</b> 11	<b>Número de HU:</b> 7
<b>Nombre de la tarea:</b> Creación de un fichero externo con extensión .txt por cada lugar de interés que contenga la información de los mismos.	
<b>Tipo de tarea:</b> Implementar	<b>Tiempo estimado:</b> 5 días
<b>Fecha de inicio:</b> 12 de febrero de 2017	<b>Fecha de fin:</b> 17 de febrero de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se crean ficheros por cada lugar de interés que contengan la información, que luego será cargada por el sistema a petición del usuario.	

*Tabla 26: Tarea de ingeniería Nro.12*

Tarea de Implementación	
<b>Número:</b> 12	<b>Número de HU:</b> 7
<b>Nombre de la tarea:</b> Implementación de la carga dinámica de la información de los ficheros externos.	
<b>Tipo de tarea:</b> Implementar	<b>Tiempo estimado:</b> 5 días
<b>Fecha de inicio:</b> 17 de febrero de 2017	<b>Fecha de fin:</b> 22 de febrero de 2017

<b>Programador responsable:</b> Damian Hernández Farramola
<b>Descripción:</b> Se implementa la función correspondiente a la lectura de la información de los ficheros externos.

*Tabla 27: Tarea de ingeniería Nro.13*

Tarea de Implementación	
<b>Número:</b> 13	<b>Número de HU:</b> 7
<b>Nombre de la tarea:</b> Implementación e integración de la funcionalidad “proyectar información”.	
<b>Tipo de tarea:</b> Implementar	<b>Tiempo estimado:</b> 10 días
<b>Fecha de inicio:</b> 25 de febrero de 2017	<b>Fecha de fin:</b> 7 de marzo de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se implementa la función que muestra en la interfaz la información referente al lugar de interés seleccionado por el usuario.	

### 3.2.5 Iteración 3

El principal objetivo de la tercera iteración es desarrollar la **HU Nro. 6** Proyectar la vista panorámica seleccionada y la **HU Nro. 4:** Navegar entre las escenas de las vistas y la interfaz principal. Esta iteración aportó como resultado la versión 2.0 del producto, permitiendo al usuario navegar por la vista panorámica seleccionada y regresar a la escena principal desde dicho punto Para su implementación se trazaron las tareas que se describen a continuación:

1. Implementación para invertir las normales de una esfera.

2. Mapeo y tratamiento de la imagen panorámica en modo esférica.
3. Texturizado de la esfera invertida con la panorámica generada.
4. Implementación de la función retornar.

*Tabla 28: Tarea de ingeniería Nro.14*

Tarea de Implementación	
<b>Número:</b> 14	<b>Número de HU:</b> 6
<b>Nombre de la tarea:</b> Implementación para invertir las normales de una esfera.	
<b>Tipo de tarea:</b> Implementar	<b>Tiempo estimado:</b> 10 días
<b>Fecha de inicio:</b> 15 de marzo de 2017	<b>Fecha de fin:</b> 25 de marzo de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se implementa una función que permita invertir todas las normales de una esfera para poder asignarle la textura del mapeado de la panorámica.	

*Tabla 29: Tarea de ingeniería Nro.15*

Tarea de Implementación	
<b>Número:</b> 15	<b>Número de HU:</b> 6
<b>Nombre de la tarea:</b> Mapeo y tratamiento de la imagen panorámica en modo esférica	
<b>Tipo de tarea:</b> Diseño	<b>Tiempo estimado:</b> 3 días

<b>Fecha de inicio:</b> 25 de marzo de 2017	<b>Fecha de fin:</b> 28 de marzo de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se toman fotos panorámicas de los locales deseados y luego se editan para poder utilizarlas como texturas esféricas.	

*Tabla 30: Tarea de ingeniería Nro.16*

Tarea de Implementación	
<b>Número:</b> 16	<b>Número de HU:</b> 6
<b>Nombre de la tarea:</b> Texturizado de la esfera invertida con la panorámica generada.	
<b>Tipo de tarea:</b> Diseño	<b>Tiempo estimado:</b> 4 días
<b>Fecha de inicio:</b> 28 de marzo de 2017	<b>Fecha de fin:</b> 1 de mayo de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se realiza el texturizado de la esfera con la panorámica esférica mapeada para lograr la vista panorámica 360°x360°.	

*Tabla 31: Tarea de ingeniería Nro.17*

Tarea de Implementación	
<b>Número:</b> 17	<b>Número de HU:</b> 6
<b>Nombre de la tarea:</b> Implementación de la función retornar	
<b>Tipo de tarea:</b> Diseño	<b>Tiempo estimado:</b> 2 días

<b>Fecha de inicio:</b> 1 de mayo de 2017	<b>Fecha de fin:</b> 3 de mayo de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se implementa una función que permite al usuario regresar a la escena principal mediante un botón, una vez cargada una de las vistas del lugar de interés seleccionado.	

### 3.3 Fase de pruebas.

Las pruebas de aceptación se crean sobre la base de las HU, en cada iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una de las HU ha sido correctamente implementada.

Las pruebas de aceptación se consideran como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. En caso que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es de todo el equipo de desarrollo, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todos estén al tanto de esta información. Su objetivo es comprobar, desde la perspectiva del usuario final, el cumplimiento de las especificaciones en cuanto a las funcionalidades del producto.

A continuación, se relacionan las pruebas de aceptación realizadas a la solución propuesta:

#### 3.3.1 Pruebas de aceptación para las HU de la iteración 1

*Tabla 32: Prueba de aceptación HU1\_P1*

Caso de prueba de aceptación	
<b>Código:</b> HU1_P1	<b>Número de HU:</b> 1

<b>Nombre:</b> Mostrar menú de opciones de vistas.
<b>Descripción:</b> Prueba para la funcionalidad mostrar menú de opciones de vistas.
<b>Condiciones de ejecución:</b> N/A
<b>Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario accede al paseo virtual mediante un navegador web.</li> <li>• Selecciona el lugar de interés.</li> <li>• Se muestra el menú mediante una animación.</li> </ul>
<b>Resultado:</b>

*Tabla 33: Prueba de aceptación HU3\_P2*

Caso de prueba de aceptación	
<b>Código:</b> HU3_P2	<b>Número de HU:</b> 3
<b>Nombre:</b> Mostrar escena en 1ra persona por el interior de la edificación de interés.	
<b>Descripción:</b> Prueba para la funcionalidad proyectar escena en 1ra persona por el interior de la edificación de interés.	
<b>Condiciones de ejecución:</b> N/A	
<b>Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario accede al paseo virtual mediante un navegador web.</li> <li>• Selecciona el lugar de interés.</li> <li>• El sistema muestra el menú de opciones.</li> <li>• El usuario selecciona la opción de vista en primera persona.</li> <li>• El sistema muestra la escena de la primera persona</li> </ul>	
<b>Resultado:</b>	

Tabla 34: Prueba de aceptación HU3\_P3

Caso de prueba de aceptación	
<b>Código:HU3_P3</b>	<b>Número de HU:3</b>
<b>Nombre:</b> Navegar en 1ra persona por el interior de la edificación de interés.	
<b>Descripción:</b> Prueba para la funcionalidad navegar en 1ra persona por el interior de la edificación de interés.	
<b>Condiciones de ejecución:</b> N/A	
<b>Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario accede al paseo virtual mediante un navegador web.</li> <li>• Selecciona el lugar de interés.</li> <li>• El sistema muestra el menú de opciones.</li> <li>• El usuario selecciona la opción de vista en primera persona.</li> <li>• El sistema carga la escena de la primera persona correspondiente al lugar de interés seleccionado previamente por el usuario.</li> <li>• El usuario puede desplazarse por la escena haciendo uso de las teclas A, S, D, W, ←, ↑, →, ↓ o mediante los botones que brinda la interfaz</li> </ul>	
<b>Resultado:</b>	

Tabla 35: Prueba de aceptación HU5\_P4

Caso de prueba de aceptación	
<b>Código:HU5_P4</b>	<b>Número de HU:5</b>
<b>Nombre:</b> Mostrar menú desplegable de las posibles vistas panorámicas del lugar de interés	

<b>Descripción:</b> Prueba para la funcionalidad mostrar menú desplegable de las posibles vistas panorámicas del lugar de interés.
<b>Condiciones de ejecución:</b> N/A
<b>Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario accede al paseo virtual mediante un navegador web.</li> <li>• Selecciona el lugar de interés.</li> <li>• El sistema muestra el menú de opciones.</li> <li>• El usuario selecciona la opción vistas panorámicas</li> <li>• El sistema muestra un menú desplegable con las opciones de vistas panorámicas.</li> </ul>
<b>Resultado:</b>

### 3.3.2 Pruebas de aceptación para las HU de la iteración 2

*Tabla 38: Prueba de aceptación HU2\_P5*

<b>Caso de prueba de aceptación</b>	
<b>Código: HU2_P5</b>	<b>Número de HU: 2</b>
<b>Nombre:</b> Mostrar información de los lugares de interés	
<b>Descripción:</b> Prueba para la funcionalidad mostrar información de los lugares de interés.	
<b>Condiciones de ejecución:</b> N/A	



<p><b>Pasos de ejecución:</b></p> <ul style="list-style-type: none"> <li>• El usuario accede al paseo virtual mediante un navegador web.</li> <li>• Selecciona el lugar de interés.</li> <li>• El sistema muestra el menú de opciones.</li> <li>• El usuario selecciona la opción información</li> <li>• El sistema carga la información desde un archivo externo.</li> <li>• El sistema muestra la información referente al lugar de interés en un panel.</li> </ul>
<p><b>Resultado:</b></p>

*Tabla 38: Prueba de aceptación HU7\_P6*

Caso de prueba de aceptación	
<b>Código:HU7_P6</b>	<b>Número de HU:7</b>
<p><b>Nombre:</b> Cargar la información de los lugares de interés desde un archivo externo con extensión .txt.</p>	
<p><b>Descripción:</b> Prueba para la funcionalidad cargar la información de los lugares de interés desde un archivo externo con extensión .txt</p>	
<p><b>Condiciones de ejecución: N/A</b></p>	
<p><b>Pasos de ejecución:</b></p> <ul style="list-style-type: none"> <li>• El usuario accede al paseo virtual mediante un navegador web.</li> <li>• Selecciona el lugar de interés.</li> <li>• El sistema muestra el menú de opciones.</li> <li>• El usuario selecciona la opción información</li> <li>• El sistema carga la información desde un archivo externo.</li> </ul>	
<p><b>Resultado:</b></p>	

### 3.3.3 Pruebas de aceptación para las HU de la iteración 3

Tabla 37: Prueba de aceptación HU6\_P7.

Caso de prueba de aceptación	
<b>Código:HU6_P7</b>	<b>Número de HU:6</b>
<b>Nombre:</b> Cargar la vista panorámica seleccionada.	
<b>Descripción:</b> Prueba para la funcionalidad navegar en la vista panorámica seleccionada.	
<b>Condiciones de ejecución:</b> N/A	
<b>Pasos de ejecución:</b> <ul style="list-style-type: none"><li>• El usuario accede al paseo virtual mediante un navegador web.</li><li>• Selecciona el lugar de interés.</li><li>• El sistema muestra el menú de opciones.</li><li>• El usuario selecciona la opción vistas panorámicas</li><li>• El sistema muestra un menú desplegable con las opciones de vistas panorámicas.</li><li>• El usuario selecciona la vista panorámica deseada.</li><li>• El sistema carga la escena panorámica seleccionada</li></ul>	
<b>Resultado:</b>	

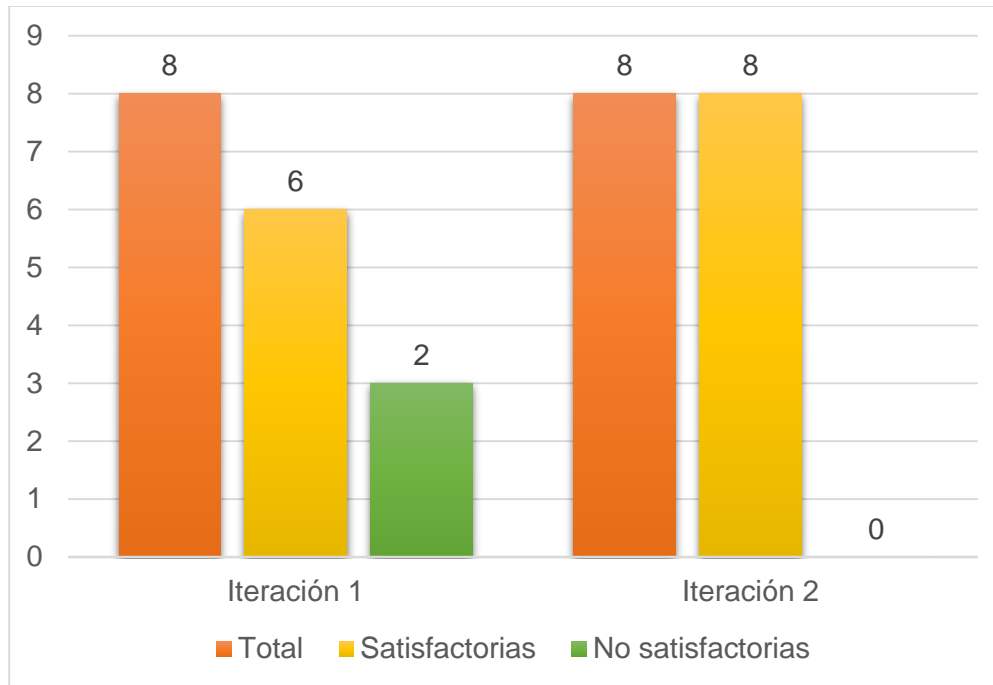
Tabla 38: Prueba de aceptación HU6\_P8

Caso de prueba de aceptación	
<b>Código:HU6_P8</b>	<b>Número de HU:6</b>
<b>Nombre:</b> Navegar en la vista panorámica seleccionada.	

<b>Descripción:</b> Prueba para la funcionalidad navegar en la vista panorámica seleccionada.
<b>Condiciones de ejecución:</b> N/A
<b>Pasos de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario accede al paseo virtual mediante un navegador web.</li> <li>• Selecciona el lugar de interés.</li> <li>• El sistema muestra el menú de opciones.</li> <li>• El usuario selecciona la opción vistas panorámicas</li> <li>• El sistema muestra un menú desplegable con las opciones de vistas panorámicas.</li> <li>• El usuario selecciona la vista panorámica deseada.</li> <li>• El sistema carga la escena panorámica seleccionada</li> <li>• El usuario puede navegar en la escena haciendo uso del mouse.</li> </ul>
<b>Resultado:</b>

### **Resultados de las pruebas de aceptación.**

Al realizar las pruebas de aceptación, en una primera iteración para un total de 8 pruebas, 6 fueron satisfactorias y 2 no satisfactorias; los errores encontrados en esta iteración fueron corregidos y en una segunda iteración, para un total de 8 pruebas 8 fueron satisfactorias. Lo cual pudo constatar la aceptación por parte del cliente de la solución desarrollada (Figura 14). El siguiente gráfico muestra el comportamiento de las pruebas antes descrito:



*Figura 14: Resultados de las pruebas de aceptación.*

### 3.4 Conclusiones parciales

En el presente capítulo se definieron las tareas de ingeniería a desarrollar, pertenecientes a cada HU en las tres iteraciones planificadas. Se implementaron además las funcionalidades definidas y se realizaron las pruebas de aceptación. Adicionalmente se concluye que:

- El uso de Unity como motor gráfico, simplificó considerablemente la exportación para la Web de la solución propuesta.
- Se realizaron dos iteraciones de pruebas y se solucionaron todas las no conformidades generadas. Obteniéndose la aceptación por parte del cliente.
- Las funcionalidades propuestas se integraron satisfactoriamente la PVW existente, adicionando nuevas formas de interacción.

## CONCLUSIONES

Con la realización de este trabajo se definieron e implementaron funcionalidades estándar para el desarrollo de paseos virtuales web, que favorecen la interacción del usuario y facilitan su mantenimiento, dando cumplimiento al objetivo propuesto al inicio de la investigación. Además, se arribó a las siguientes conclusiones:

- La definición de las funcionalidades estándar para la realización de paseos virtuales web, sirve de apoyo para el desarrollo de futuros proyectos. Además, dicho estándar favorece la inmersión y la interacción de los usuarios con los paseos virtuales.
- Las funcionalidades implementadas adicionaron nuevas formas de interacción al PVW y facilitaron la realización de modificaciones de la información de los sitios de interés, luego de realizado el despliegue.
- La integración de las vistas interiores (primera persona y panorámicas), permiten explorar las edificaciones de modo más inmersivo.

## RECOMENDACIONES

Se recomienda para futuras implementaciones: incorporar funcionalidades que permitan cargar dinámicamente modelos tridimensionales que representen lugares de interés, lo que favorecería la extensibilidad de los paseos virtuales generados.

## REFERENCIAS

- [.NET]. 2015. System.OutOfMemoryException. [En línea] 18 de 05 de 2015. <https://sparraguerra.wordpress.com/2015/05/18/un-poquito-de-historia-de-c/>.
- Baeza, Tomás Alemañ. 2015.** *Desarrollo de un videojuego para móviles con Unity*. Alicante : Universidad de Alicante, 2015.
- Carmona, Juan García. 2012.** Buenas prácticas hacia el éxito en el desarrollo del software. 2012.
- Chorafas, Dimitris y Steinmann, Heinrich. 2009.** *Realidad virtual: aplicaciones prácticas en los negocios y la industria*. Madrid : Hispanoamericana S.A., 2009.
- Dale, A. 2009.** *Comuncations & Multimedia Technology*. Canadá : Digital Overdrive, 2009.
- EcuRed.** Visita virtual - EcuRed. [En línea] [Citado el: 13 de noviembre de 2016.] [http://www.ecured.cu/index.php/Visita\\_virtual..](http://www.ecured.cu/index.php/Visita_virtual..)
- Gamma, Erich. 1995.** *Design Patterns. Elements of Reusable Object-Oriented Software*. Wesley : Addison, 1995.
- Génova, Gonzalo. 2008.** *Conceptos basicos del modelado*. Madrid : s.n., 2008.
- Gubern, Roman. 2009.** *Del bisonte a la realidad virtual*. Barcelona : Anagrama, 2009.
- Lenguajes de Programación 2009. *Lenguajes de Programación 2009*. [En línea] <http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml..>
- Leon, Mijail. 2013.** Diagrama de Clases. Turmero : s.n., 2013.
- LLC, Tangient. 2013.** [En línea] 10 de marzo de 2013. [Citado el: 14 de enero de 2017.] <http://programacion-extrema.wikispaces.com/5.+Ciclo+de+vida+y+fases>.
- López, Gonzalo. 2011.** Informática XIV Convención y Feria Internacional 2011. *Informática XIV Convención y Feria Internacional 2011*. [En línea] 10 de Febrero de 2011. [www.informaticahabana.cu..](http://www.informaticahabana.cu..)
- Maldonado, Tomas. 2008.** *Lo real y lo virtual*. Barcelona : Gedisa, 2008.
- Mestras, Juan Pavón. 2004.** *Patrones de diseño orientado a objetos*. Madrid : s.n., 2004.
- Meza, Mirna. 2011.** Herramientas Case. *Herramientas Case*. [En línea] 2 de abril de 2011. <http://fds-herramientascase.blogspot.com/>.

- Microsoft. 2015.** Developer Network. Revisiones de código y estándares de codificación. [En línea] 2015. <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>).
- Myers, Glenford J.** *The art of software testing*. s.l. : 2da edicion.
- Oswaldo Castillo, Hector Sevilla y Daniel Figueroa. 2016.** *Programacion Extrema*. [En línea] tripod, 2016. <http://programacionextrema.tripod.com/fases.htm..>
- Pressman, Roger. 2010.** *Ingeniería de Software. Un enfoque práctico*. s.l. : Septima Edicion , 2010.
- Roberth G. Figueroa, Camilo J. Solis, Armando A. Cabrera. 2016.** *Metodologías tradicionales Vs metodologías ágiles*. Loja : s.n., 2016.
- Ryan, Marie Lauren. 2009.** *Narrative as Virtual Reality: Immersion and Interactive in Literature and Electronic Media*. New York: : Baltimore and London, 2009.
- Sanchez, Alonso. 2013.** [En línea] 21 de Enero de 2013. [https://prezi.com/7wmx8\\_d6ertl/entorno-desarrollo-integrado-ide/](https://prezi.com/7wmx8_d6ertl/entorno-desarrollo-integrado-ide/).
- Sanchez, T Rodrigo. 2014.** *Metodología de desarrollo para la Actividad productiva de la UCI*. 2014.
- Scribd, Scribd Inc. 2014.** [En línea] 20 de abril de 2014. <http://es.scribd.com/doc/45010504/Tarjetas-CRC>.
- Solórzano, J. L. 2012.** *Herramientas de Modelado 3D*. México : Trillas : s.n., 2012.
- Ton Rosendaal, Stefano Selleri. 2009.** *La Suit Abierta De Creacion 3D*. 2009.
- Ulldemolins, Álvaro. 2011.** *Recorridos virtuales*. Catalunya : UOC, 2011.
- Unity. 2017.** Unity Documentation. *Unity Documentation*. [En línea] 19 de abril de 2017. <https://docs.unity3d.com>.



## ANEXOS

### Tareas de ingenierías para la iteración Nro. 1

*Tabla 20: Tarea de ingeniería Nro.6*

Tarea de Implementación	
<b>Número:6</b>	<b>Numero de HU:5</b>
<b>Nombre de la tarea:</b> Implementación e integración de la funcionalidad del menú desplegable al paseo virtual.	
<b>Tipo de tarea:</b> Implementación	<b>Tiempo estimado:</b> 2 días
<b>Fecha de inicio:</b> 2 de enero de 2017	<b>Fecha de fin:</b> 4 de enero de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se implementa la funcionalidad del menú desplegable que permite al usuario seleccionar una opción de vista panorámica referente al lugar de interés previamente seleccionado	

*Tabla 21: Tarea de ingeniería Nro.7*

Tarea de Implementación	
<b>Número:7</b>	<b>Número de HU:3</b>
<b>Nombre de la tarea:</b> Modelado el interior de un lugar de interés, en este caso se escogió la Biblioteca.	
<b>Tipo de tarea:</b> Diseño	<b>Tiempo estimado:</b> 5 días
<b>Fecha de inicio:</b> 4 de enero de 2017	<b>Fecha de fin:</b> 9 de enero de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	

**Descripción:**

Se modela el interior del lugar de interés "Biblioteca" que permita al usuario experimentar el paseo virtual en primera persona.

*Tabla 22: Tarea de ingeniería Nro.8*

Tarea de Implementación	
<b>Número:8</b>	<b>Número de HU:3</b>
<b>Nombre de la tarea:</b> Diseño de los botones para la navegación en 1ra persona.	
<b>Tipo de tarea:</b> Diseño	<b>Tiempo estimado:</b> 2 días
<b>Fecha de inicio:</b> 9 de enero de 2017	<b>Fecha de fin:</b> 11 de enero de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se diseñan los botones que permiten al usuario navegar por la escena en 1ra persona.	

*Tabla 23: Tarea de ingeniería Nro.9*

Tarea de Implementación	
<b>Número:9</b>	<b>Número de HU:3</b>
<b>Nombre de la tarea:</b> Implementación del controlador de 1ra persona.	
<b>Tipo de tarea:</b> Implementar	<b>Tiempo estimado:</b> 5 días

<b>Fecha de inicio:</b> 11 de enero de 2017	<b>Fecha de fin:</b> 16 de enero de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se implementan las funciones necesarias para permitir el uso del controlador de 1ra persona.	

*Tabla 24: Tarea de ingeniería Nro.10*

Tarea de Implementación	
<b>Número:</b> 10	<b>Número de HU:</b> 3
<b>Nombre de la tarea:</b> Implementación de la funcionalidad de los botones para el controlador de 1ra persona.	
<b>Tipo de tarea:</b> Implementar	<b>Tiempo estimado:</b> 3 días
<b>Fecha de inicio:</b> 16 de enero de 2017	<b>Fecha de fin:</b> 19 de enero de 2017
<b>Programador responsable:</b> Damian Hernández Farramola	
<b>Descripción:</b> Se implementan las funciones de los botones con los que interactúa el usuario.	

**Plantilla para la confección de las tarjetas CRC.**

*Tabla 10: Plantilla de las tarjetas CRC*

Tarjeta CRC
<b>Clase:</b> "Nombre de la clase que se está modelando".

**Responsabilidad:** Es una descripción de alto nivel del propósito de la clase.

**Colaboración:** Indica con cuales otras clases se requiere relación para cumplir la responsabilidad.