

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN MUSEO
VIRTUAL EN 3D PARA LA FUNDACIÓN GUAYASAMÍN,
UTILIZANDO EL FRAMEWORK XNA PARA APLICACIONES
DESKTOP MULTIMEDIA

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR:

LUIS MIGUEL CÓNDOR GUACHAMÍN

SANGOLQUÍ, ABRIL – 2012

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. LUIS MIGUEL CÓNDROR GUACHAMÍN como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

13 de Abril del 2012

ING. CESAR VILLACÍS

DEDICATORIA

Al terminar esta etapa tan importante de mi vida, dedico el presente trabajo a mis padres, ya que sin su apoyo constante, no hubiese alcanzado a culminar esta meta tan anhelada.

Luis Miguel Cóndor Guachamín

AGRADECIMIENTOS

Mi eterna gratitud a quienes supieron apoyarme en todo momento, de manera especial a mis profesores, mi familia y sobre todo mis padres.

De manera especial, un enorme agradecimiento a mis compañeros de toda la vida: Soledad Hernández (“SolKool”) y Alfredo Cárdenas (“7U15MK”), amigos inigualables que supieron acompañarme en este arduo e intrincado viaje que se llama vida.

Y un personal agradecimiento a la mujer más especial que existe en mi vida: Alejandra Cuadros, quien lleno mi corazón de los más puros sentimiento de afecto, cariño y amor, y quien es la inspiración de mis proyectos presentes y futuros.

Luis Miguel Cóndor Guachamín

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN.....	i
DEDICATORIA.....	ii
AGRADECIMIENTOS	iii
ÍNDICE DE CONTENIDOS	iv
LISTADO DE TABLAS	vii
LISTADO DE FIGURAS	viii
GLOSARIO.....	ix
ACRÓNIMOS	x
RESUMEN.....	1
CAPÍTULO 1.....	2
1. INTRODUCCIÓN.....	2
1.1 PRÓLOGO	2
1.2 PLANTEAMIENTO DEL PROBLEMA.....	3
1.3 JUSTIFICACIÓN	4
1.4 OBJETIVOS	5
1.4.1 Objetivo General.....	5
1.4.2 Objetivos Específicos	5
1.5 ALCANCE.....	6
CAPÍTULO 2.....	8
2. INTRODUCCIÓN.....	8
2.1 GRÁFICOS 3D GENERADOS POR COMPUTADOR	8
2.1.1 Breve reseña histórica de la generación de gráficos 3D por computador	8
2.1.2 Geometría en 3D por computador.....	12
2.1.2.1 Conceptos Básicos.....	12
2.1.2.2 Proyección de Perspectiva.....	13
2.1.2.3 Conceptos Sobre Gráficos en 3D.....	15
2.1.2.4 Software para diseño en 3D.....	16
2.2 METODOLOGÍA PARA EL DESARROLLO DEL APLICATIVO 3D.....	19
2.2.1 Ingeniería de Software aplicada al desarrollo de aplicativos	19
2.2.1.1 Capas de Ingeniería de Software.....	20
2.2.2 Lenguaje de Modelado UML	21
2.2.2.1 Diagramas Recomendados	22
2.2.3 El Proceso Scrum.....	24
2.2.3.1 Beneficios de Scrum.....	24
2.2.3.2 El Proceso	25
2.2.3.3 Roles.....	26
2.2.3.4 Actividades	27

2.2.3.5	Artefactos	28
2.2.4	Framework XNA 3.1	29
2.2.4.1	Aspectos Generales sobre XNA.....	30
CAPÍTULO 3.....		32
3.	ANÁLISIS, DISEÑO Y DESARROLLO	32
3.1	CARACTERÍSTICAS DE SCRUM	32
3.1.1	Roles.....	32
3.1.1.1	Roles Principales.....	32
3.1.1.2	Roles Secundarios	34
3.1.2	Artefactos de Scrum	35
3.1.3	Iteraciones de Scrum.....	36
3.2	ITERACIÓN I: DISEÑO DEL MODELO EN 3D.....	37
3.2.1	Resumen	37
3.2.2	Reunión de Planificación del Sprint (Sprint Planning Meeting).....	37
3.2.3	Pila de la Iteración (Sprint Backlog)	38
3.2.4	Tareas.....	41
3.2.4.1	ÍTEM 1: Tomar las medidas físicas del museo.	41
3.2.4.2	ÍTEM 2: Fotografiar las obras en exposición.....	42
3.2.4.3	ÍTEM 3: Modelar el museo en 3ds Max.....	44
3.2.4.4	ÍTEM 4: Determinar la iluminación a usar en el modelo.	49
3.2.4.5	ÍTEM 5: Realizar un video del modelo del museo.....	50
3.2.5	Gráfica de Trabajo Pendiente	51
3.2.6	Reunión de Revisión de la Iteración (Sprint Review Meeting).....	52
3.3	ITERACIÓN II: DESARROLLO DEL APLICATIVO SOFTWARE	53
3.3.1	Resumen	53
3.3.2	Reunión de Planificación del Sprint (Sprint Planning Meeting).....	53
3.3.3	Pila de la Iteración (Sprint Backlog)	54
3.3.4	Diseño.....	56
3.3.4.1	Introducción	56
3.3.4.2	Especificación de requisitos de software	58
3.3.4.3	Casos de Uso	62
3.3.4.4	Diagramas	68
3.3.5	Tareas.....	82
3.3.5.1	ÍTEM 6: Configurar las herramientas de XNA Game Studio 3.1.....	82
3.3.5.2	ÍTEM 7: Desarrollar el recorrido virtual por el museo.....	83
3.3.5.3	ÍTEM 8: Generar el instalador del aplicativo.	85
3.3.5.4	ÍTEM 9: Documentar los manuales de usuario.	91
3.3.6	Documentos generados	91
3.3.7	Gráfica de trabajo pendiente	91

3.3.8	Reunión de revisión de la iteración (Sprint Review Meeting).....	92
3.4	ITERACIÓN III: COMPLEMENTOS MULTIMEDIA.....	94
3.4.1	Resumen	94
3.4.2	Reunión de Planificación del Sprint (Sprint Planning Meeting).....	94
3.4.3	Pila de la Iteración (Sprint Backlog)	95
3.4.4	Tareas.....	96
3.4.4.1	ÍTEM 10: Documentar la información de las obras y exposiciones.	96
3.4.4.2	ÍTEM 11: Desarrollar los diferentes estados del aplicativo.	96
3.4.4.3	ÍTEM 12: Integrar la base documental en el aplicativo.	97
3.4.4.4	ÍTEM 13: Compilar toda la información en un medio óptico.	98
3.4.5	Documentos Generados.....	99
3.4.6	Gráfica de Trabajo Pendiente	99
3.4.7	Reunión de Revisión de la Iteración (Sprint Review Meeting).....	100
CAPÍTULO 4.....		101
4.	CONCLUSIONES Y RECOMENDACIONES.....	101
4.1	CONCLUSIONES.....	101
4.2	RECOMENDACIONES	103
BIBLIOGRAFÍA.....		104
BIOGRAFÍA.....		105

LISTADO DE TABLAS

Tabla 3.1: Pila del producto.....	35
Tabla 3.2: Pila de la Iteración I - Primera parte.....	39
Tabla 3.3: Pila de la Iteración I - Segunda parte.....	40
Tabla 3.4: Pila de la Iteración II - Primera parte.....	54
Tabla 3.5: Pila de la Iteración II - Segunda parte.....	55
Tabla 3.6: Requisitos funcionales del sistema.	61
Tabla 3.7: Métodos y atributos de la clase Camera.....	73
Tabla 3.8: Atributos de la clase Game1.	74
Tabla 3.9: Métodos de la clase Game1.	75
Tabla 3.10: Métodos de la clase MenuInteraction.	76
Tabla 3.11: Métodos y atributos de la clase MuseumData.	77
Tabla 3.12: Métodos y atributos de la clase Movement.....	78
Tabla 3.13: Pruebas de recuperación.	87
Tabla 3.14: Pruebas de resistencia.....	88
Tabla 3.15: Pila de la Iteración III.....	95

LISTADO DE FIGURAS

Figura 2.1: Sketchpad de Ivan Sutherland.....	9
Figura 2.2: Tetera renderizada en 3D.....	10
Figura 2.3: Imagen 2D (arriba), imagen 3D (abajo).....	12
Figura 2.4: Proyección Isométrica.....	13
Figura 2.5: Proyeccion Ortogonal.....	14
Figura 2.6: Google SketchUp.....	16
Figura 2.7: Entorno de trabajo de Blender.....	17
Figura 2.8: Pantalla inicial de 3ds Max 2010.....	18
Figura 2.9: Capas de Ingeniería de Software.....	20
Figura 2.10: Método utilizado por Scrum.....	21
Figura 2.11: El proceso Scrum.....	25
Figura 2.12: Capas de XNA.....	31
Figura 3.1: Fotografía de la parte exterior del museo.....	41
Figura 3.2: Redimensión de las imágenes en Photoshop.....	43
Figura 3.3: Formato de enumeración de imágenes.....	44
Figura 3.4: Primer piso del modelo.....	45
Figura 3.5: Piso inferior del modelo.....	46
Figura 3.6: Lobby del museo.....	47
Figura 3.7: Exterior del museo.....	48
Figura 3.8: Esqueleto del museo en 3D.....	49
Figura 3.9: Vista del modelo con una cámara.....	50
Figura 3.10: Gráfico de Trabajo Pendiente - Iteración I.....	51
Figura 3.11: Casos de Uso.....	62
Figura 3.12: Clases de Aplicativo.....	68
Figura 3.13: Métodos y atributos de la clase Game1.....	69
Figura 3.14: Métodos y atributos de la clase Camera.....	70
Figura 3.15: Métodos de la clase MenuInteraction.....	71
Figura 3.16: Métodos y atributos de la clase MuseumData.....	71
Figura 3.17: Métodos y atributos de la clase Movement.....	72
Figura 3.18: Diagrama del estado inicial.....	79
Figura 3.19: Diagrama del estado de la ejecución del recorrido.....	80
Figura 3.20: Diagrama de actividad.....	81
Figura 3.21: Modelo en 3D de prueba.....	82
Figura 3.22: Modelo del museo sin las obras.....	83
Figura 3.23: Obras cargadas dentro del modelo.....	85
Figura 3.24: Archivos generados.....	86
Figura 3.25: Rendimiento del equipo sin ejecutar la aplicación.....	89
Figura 3.26: Rendimiento del equipo mientras se ejecuta la aplicación.....	90
Figura 3.27: Gráfico de Trabajo Pendiente - Iteración II.....	92
Figura 3.28: Captura de pantalla de la base documental.....	98
Figura 3.29: Gráfico de Trabajo Pendiente - Iteración III.....	99

GLOSARIO

3ds Max:	Tridimensional.
Aplicativo:	Software o programa de computador que le permiten al usuario llevar a cabo las tareas que desea ejecutar.
Código Fuente:	Conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar un programa.
C# .NET:	Lenguaje de programación orientado a objetos desarrollado por Microsoft.
Framework:	Esquema para el desarrollo y/o la implementación de una aplicación.
Modelado:	Creación y modificación de objetos en tres dimensiones.
Multimedia:	Objeto o sistema que utiliza múltiples medios de expresión (físicos o digitales) para presentar o comunicar información. Los medios pueden ser variados: texto, imágenes, animación, sonido, video, etc.
Parámetro:	Variable que puede ser recibida por una rutina o subrutina, alterando su comportamiento en tiempo de ejecución.
Renderizar:	Proceso de generar una imagen desde un modelo.
Stakeholders:	Cualquier persona o entidad que es afectada por las actividades de una organización.
Virtual:	Algo que tiene existencia aparente y no real, término utilizado para designar funciones o dispositivos simulados.

ACRÓNIMOS

- 3D** Tridimensional.
- CD** Disco Compacto.
- IDE** Entorno de Desarrollo Integrado.
- SDK** Kit de Desarrollo De Software.
- UML** Lenguaje Unificado de Modelado.
- XNA** Nueva Arquitectura Xbox.

RESUMEN

La Capilla del Hombre, museo concebido por el maestro Oswaldo Guayasamín, constituye la inspiración y el punto de partida para el presente proyecto de tesis, mediante el cual se analiza el aspecto teórico y práctico de la implementación de un recorrido virtual e interactivo por el mencionado establecimiento.

El levantamiento de una edificación en un entorno virtual conlleva la utilización de herramientas software, tanto para el modelado de la estructura en un ambiente virtual, como para la implementación de la interacción entre el usuario y el modelo creado.

El programa Autodesk 3ds Max 2010 fue la herramienta idónea para el modelado virtual del museo, modelo que posteriormente fue posible recorrerlo e interactuar con el mismo mediante la codificación de un programa utilizando el Framework XNA y el entorno de desarrollo integrado Visual Studio 2008.

La naturaleza propia de un museo requiere de la exposición de obras artísticas, debido a lo cual fue necesario digitalizar las pinturas y esculturas para su posterior visualización dentro del recorrido virtual. Todo dentro de una interfaz sencilla e intuitiva de manejar debido a que el aplicativo final será usado con fines educativos y publicitarios para una mayor difusión del museo tanto en el ámbito local como internacional.

CAPÍTULO 1

1. INTRODUCCIÓN

1.1 PRÓLOGO

La tecnología avanza a pasos agigantados, aunque en el pasado se veía a los computadores solo como equipos empresariales, ahora se evidencia como éstos dejaron de ser simples máquinas procesadoras de números y ofrecen al usuario experiencias en vídeo voz y datos, creando innumerables opciones a desarrollarse en el ámbito multimedia. Esto, junto con la acelerada inserción de los equipos computacionales tanto en los hogares como en las instituciones permite que todo tipo de contenido pueda ser distribuido a más de un billón de potenciales usuarios¹.

Si bien en un inicio los motores de juegos fueron diseñados con el propósito de simplificar el nivel de abstracción al momento de desarrollar un videojuego, ahora se han tornado en herramientas que permite la creación de contenido interactivo para el usuario. Estas herramientas no solo permiten la creación de videojuegos, sino que además pueden ser vinculadas al ámbito médico, con la creación de máquinas de entrenamiento para cirugías y procedimientos médicos. Además se los puede visualizar en el ambiente arquitectónico, para la modelación de edificaciones a ser construidas o también al poder ser usadas para desarrollar simuladores de vuelo para el entrenamiento de los futuros aviadores, y obviamente para el efecto de esta tesis, se pueden usar con el fin de producir una réplica virtual de un museo en el cual no solo el usuario recorre el establecimiento, sino que además interactúa con los

¹ Según estudios estadísticos de Gartner Inc. realizados en el año 2008.

elementos que encuentra en su visita para tener una mayor información y una mayor profundización en los diferentes elementos que componen al museo.

Actualmente está en auge el diseño y la construcción de museos virtuales por todo el mundo, como un medio para promocionar las diferentes exposiciones que están presentes, así por ejemplo el Museo Louvre de Paris posee una de las más grandes exposiciones virtuales. También existen instituciones como el Museo de Historia Mexicana, el Museo Metropolitano de New York, el Museo de Prado en España, entre otros, que optan por utilizar un medio virtual para su difusión.

1.2 PLANTEAMIENTO DEL PROBLEMA

Mediante la elaboración del proyecto de tesis se propone solucionar los siguientes problemas:

El museo La Capilla del Hombre, posee exposiciones de: pintura, fotografía y escultura en forma tradicional y solo disponible mediante la visita física al establecimiento. El material multimedia obtenido permitirá que muchas personas que viven tanto en el exterior como en las diferentes provincias de nuestro país conozcan a detalle de la amplia gama de obras artísticas que se encuentran alojadas en el museo.

La información disponible tanto del establecimiento como de cada una de las obras artísticas es escasa, por lo cual dicha información será incluida en el aplicativo a fin de dar al usuario la posibilidad de profundizar más en los detalles de cada una de las obras.

El proyecto además permite una mayor accesibilidad para personas que no dispongan del tiempo para visitar físicamente el establecimiento. Al almacenar el museo virtual en un sitio web se da al usuario la posibilidad de descargarlo y ejecutarlo cuando ellos dispongan del tiempo necesario. Permite una difusión más extensa de las obras de gran valor cultural del maestro Oswaldo Guayasamín.

1.3 JUSTIFICACIÓN

El presente proyecto es necesario para la Fundación Guayasamín, específicamente el museo La Capilla del Hombre, debido a que se logrará la creación de un medio digital de difusión de las diferentes exposiciones existentes en dicho museo, dando lugar a una exposición a nivel mundial de las diferentes obras; de igual modo, complementariamente es conveniente para las diferentes instituciones vinculadas a la Fundación Guayasamín, debido a que serán beneficiadas de la publicidad generada por el proyecto mencionado.

El uso del Framework XNA permitirá la creación de medios de interacción más completos dentro del entorno virtual creado, al permitir recorrer los diferentes espacios del museo ya sea en un modo automático o mediante el control propio del usuario de las direcciones de movimiento, además de visualizar en detalle la información de cada una de las obras artísticas. A diferencia de los motores de juego como Unreal Tech, ID Tech, Cry Tech, etc. XNA permite trabajar a un nivel más bajo, dando mayor control y enfocándose en el tema de la programación y codificación del aplicativo.

Una de las ventajas principales del uso del lenguaje de programación C# para la realización del proyecto es su integración robusta con el Framework XNA. Además, brinda ventajas al momento de realizar la programación, como por ejemplo: tiempo de compilación menor, manejo de memoria automático, facilidad al momento de depurar, facilidad para crear interfaces de usuario, evita errores producidos por librerías y definiciones, entre otros.

El modelo en 3D del museo será realizado utilizando la herramienta Autodesk 3ds Max Studio 2010, la cual tiene un mayor enfoque a la realización de estructuras y edificaciones, a diferencia de otras herramientas como Autodesk Maya, cuyas herramientas se enfocan mayormente en el modelado de personas, animales y afines; o herramientas como Blender, que se enfocan en la realización de videos y animaciones.

1.4 OBJETIVOS

1.4.1 Objetivo General

- Desarrollar un Museo Virtual en 3D para la Fundación Guayasamín, utilizando el Framework XNA para aplicaciones desktop multimedia.

1.4.2 Objetivos Específicos

- Realizar el diseño y desarrollo de la aplicación del museo virtual con el Framework XNA aplicando la metodología Scrum.
- Desarrollar el modelo 3D de la aplicación con la herramienta Autodesk 3ds Max 2010.

- Implementar el recorrido virtual del museo mediante el uso del Framework XNA y C#.NET como el lenguaje de programación.
- Proporcionar un recurso de bajo costo para realizar una visita al museo de la Capilla del hombre. La aplicación final será gratuita para el usuario final.
- Proporcionar una herramienta para visitar virtualmente el museo de la Capilla del Hombre desde cualquier lugar del país o del mundo.

1.5 ALCANCE

Al realizar el presente proyecto de tesis, se toma como punto de partida la metodología Scrum, como un marco de referencia para la definición de los diferentes parámetros del proyecto y vinculada a UML, mediante el uso de los diferentes diagramas propuestos por este lenguaje de modelado. Los cuales abarcan todo el proceso de desarrollo del software que permite la creación del aplicativo desktop, mediante el cual se realiza el recorrido virtual.

Debido a la naturaleza del proyecto, es necesario realizar el diseño del modelo virtual del museo de la Capilla del Hombre en un ambiente 3D usando diferentes herramientas de software orientadas específicamente a este fin, dando una mayor gama de opciones para crear un ambiente virtual más realista. Además, se plantea la creación de un aplicativo multimedia el cual será distribuido en diferentes medios de almacenamiento, a fin de que el usuario pueda utilizar el producto final en su propio computador. Todas estas consideraciones permiten definir los puntos clave de este proyecto de tesis, los cuales son:

- Aplicar la Metodología Scrum con UML, para la definición de los parámetros del proyecto. Realización de los diagramas de UML y definición de las iteraciones definidas por Scrum para el proyecto.
- Diseñar el Museo Virtual de la Capilla del Hombre de la Fundación Guayasamín en un ambiente 3D, el cual incluye los dos niveles del museo, el patio central alrededor del edificio y el exterior.
- Desarrollar el aplicativo desktop para el recorrido virtual, el cual unificará el modelo virtual con el Framework XNA y permitirá el recorrido por las diferentes salas del museo, además de mostrar la información de cada una de las obras en exposición.
- Desarrollar un CD multimedia con el aplicativo desktop, que contendrá los instaladores para el Framework .NET 3.5, instaladores del aplicativo desktop, base documental de la información de las exposiciones del museo y el manual de usuario.

CAPÍTULO 2

2. INTRODUCCIÓN

2.1 GRÁFICOS 3D GENERADOS POR COMPUTADOR

2.1.1 Breve reseña histórica de la generación de gráficos 3D por computador

Como prefacio al desarrollo del presente proyecto es necesaria una breve introducción al ámbito en el cual se ha trabajado, puntualmente a la historia y al estado actual de los medios y métodos para la generación de gráficos 3D en un computador.

La computación gráfica es una disciplina que abarca la producción de imágenes utilizando un computador, tarea realizada mediante el uso de: técnicas de modelado, creación, manipulación y almacenamiento de objetos geométricos: y de renderización, para convertir una escena en imagen, o también refiriéndose a la transformación, rasterización, sombreado, iluminación, y animación de una imagen (Impagliazzo, 2004).

La computación gráfica ha sido utilizada ampliamente en presentaciones, procesamiento de imágenes, diseño asistido por computador, simulación, realidad virtual, y el entretenimiento. Evolucionando desde simples imágenes de figuras básicas hasta la generación de imágenes foto realistas, de interfaces simples² para el diseño de objetos a capturas en 3D de objetos reales (Impagliazzo, 2004).

² Refiriéndose al teclado y mouse utilizados comúnmente para operar un computador.



Figura 2.1: Sketchpad de Ivan Sutherland.

El inicio de la computación gráfica se lo puede atribuir a Ivan Sutherland (Sevo, 2010), estudiante del MIT³ quien en 1961 creó un programa de dibujo por computador llamado Sketchpad, como se visualiza en la figura 2.1. El cual permitía dibujar figuras simples en la pantalla del computador, grabarlas y volverlas a usar cuando se lo requiera, utilizando un lápiz digital que poseía una celda fotoeléctrica en la punta, que se le daba la capacidad de interactuar con el computador. Los gráficos generados por aquel entonces no eran más que simples líneas, llamadas vectores, mientras que en la actualidad se usan técnicas de rasterizado para representar los gráficos por medio de píxeles.

Científicos de varias aéreas empezaron a usar la computación gráfica para generar modelos o representaciones de sus trabajos, tales como: “El flujo de un fluido viscoso”, “Propagación de una onda en una forma solida” y “vibración de un aeroplano” en el laboratorio de radiación de Lawrence.

³ Manchester Institute of Technology

No paso mucho antes de que las grandes corporaciones lograran visualizar el gran potencial y tomaran interés en la computación gráfica. TRW, Lockheed-Georgia, General Electric y Sperry Rand son algunas de las corporaciones que incursionaron en la computación gráfica a mediados de los 60's. IBM fue muy oportuna en responder a esta nueva tecnología con la fabricación de la terminal gráfica IBM 2250, la cual se convirtió en la primera computadora gráfica disponible comercialmente.

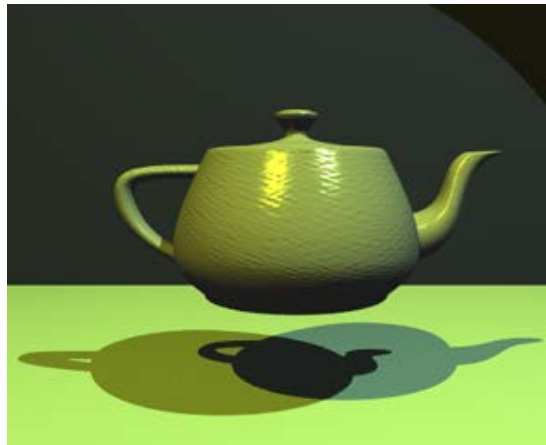


Figura 2.2: Tetera renderizada en 3D.

El primer gran avance en la computación gráfica en 3D fue creado en la Universidad de Utah por varios pioneros que propusieron el algoritmo de la superficie oculta. Algoritmo por el cual se podía representar un objeto en 3D en pantalla, determinando cuál superficie está detrás del objeto en relación a la perspectiva del observador, y procediendo a esconderlo al momento de renderizar la imagen en pantalla, como se ejemplo se puede visualizar en la Figura 2.2.

La búsqueda del realismo fue el impulso para la generación de nuevas ideas y paradigmas dentro de la renderización de imágenes, como por ejemplo el método de sombreado de un objeto propuesto por Henri Gouraud en 1971, con el cual se podía

representar una superficie curva mediante la interpolación de los colores en un polígono, dando una mayor calidad a los objetos renderizados.

En la década de los 70's el trabajo y la investigación fue orientado al realismo de los objetos renderizados, se propusieron métodos para usar patrones en las texturas de los objetos, simulando los patrones que se usan en las telas, además de la aplicación de luces y sombreado a las diferentes áreas de un objeto para poder generar una representación de un objeto en 3D (Impagliazzo, 2004). Mientras que en los años 80, el auge de los computadores personales y su costo reducido permitió un desarrollo más independiente de modelos y técnicas de generación en 3D, en parte debido a la creación del programa Polhemus, el cual fue el primer software de creación de gráficos en 3D.

Películas como "Tron"⁴ de la productora Disney usaron una amplia gama de efectos en 3D, además de "Luxo Jr." y "Tin toy" de Pixar Animation Studio. Para los años 90 el cine vio nacer a clásicos como "Jurassic Park" y "Toy Story" siendo esta última, la primera película de larga duración generada completamente por computador.

Desde juegos hasta la realidad virtual, desde sistemas ambientales caseros a aplicaciones científicas y comerciales, los gráficos por computadora han tocado cada uno de los aspectos de nuestras vidas. Las tendencias continúan y el futuro solo es incierto (Impagliazzo, 2004).

⁴ Película que trata sobre un viaje al interior de un juego de video.

2.1.2 Geometría en 3D por computador

2.1.2.1 Conceptos Básicos

El término gráficos 3D por computador hace referencia a trabajos de arte gráfico que son creados con ayuda de computadoras y software específico. El término también puede referirse al proceso de crear dichos gráficos, o al campo de estudio de técnicas y tecnología que tienen relación con los gráficos 3D.

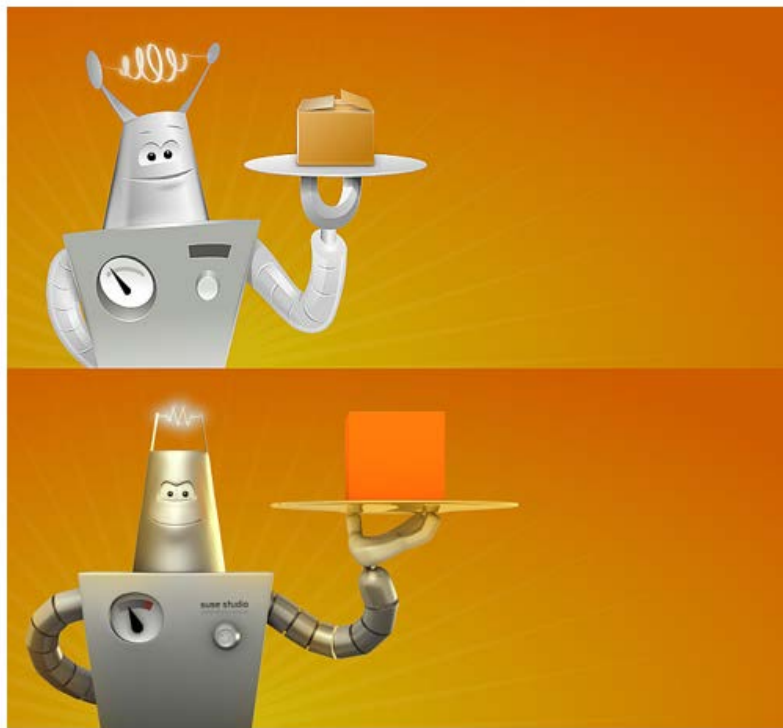


Figura 2.3: Imagen 2D (arriba), imagen 3D (abajo).

La diferencia fundamental entre una imagen generada en 3D y una generada en 2D está en la forma por la cual se la ha generado (Figura 2.3). Las imágenes en 3D son originadas mediante un proceso de cálculos matemáticos sobre entidades geométricas tridimensionales producidas en un ordenador, y cuyo propósito es

conseguir una proyección visual en dos dimensiones para ser mostrada en un medio visual⁵.

2.1.2.2 Proyección de Perspectiva

La perspectiva se define como el método de representar los objetos en la forma y la disposición con que se aparecen a la vista. También, como el conjunto de objetos que se visualizan desde el punto de vista del espectador (Conte, 2010).

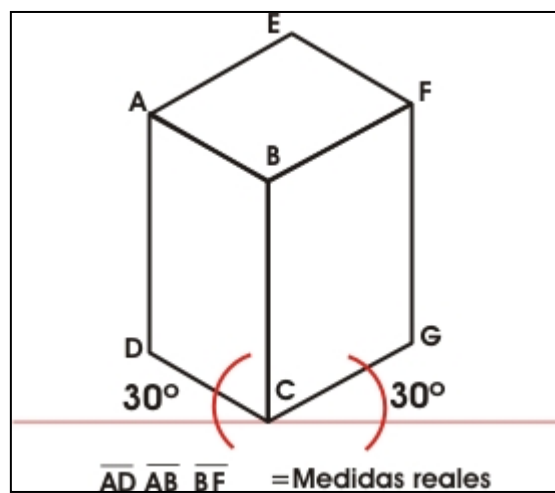


Figura 2.4: Proyección Isométrica.

Dentro del marco del proyecto actual, la perspectiva representa un papel importante a modo de permitir la representación de un objeto tridimensional en una superficie de dos dimensiones, lo cual ayuda a crear una sensación de profundidad, y de espacialidad dentro de un modelo virtual. Cabe recalcar la necesidad de usar la perspectiva isométrica (figura 2.4) para el presente proyecto, debido a su utilización dentro del software de modelado en 3D.

⁵ Por ejemplo la pantalla de un ordenador o una hoja de papel impresa.

Un modo similar de representar un objeto en dos dimensiones es la proyección ortogonal (figura 2.5), en donde los haces de líneas proyectantes son perpendiculares al plano. Cualquier objeto puede ser visualizado desde diferentes puntos de vista lo que permite determinar de manera más objetiva su estructura, conociendo mejor cada una de sus partes.

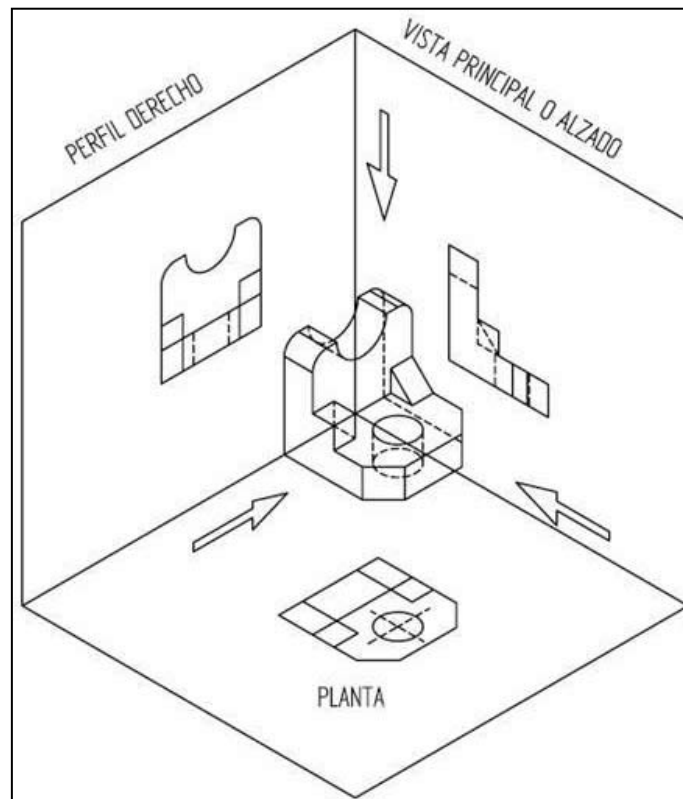


Figura 2.5: Proyección Ortogonal.

Las técnicas fundamentales utilizadas para obtener perspectivas son: controlar la variación entre los tamaños de los sujetos u objetos representados, superponiendo algunos de ellos, y colocando los que están pintados en el terreno que se representa, más abajo cuando están más cerca y más altos cuando están más lejos (Conte, 2010), todo esto es posible generarlo en una herramienta de modelado en 3D para la virtualización de un objeto y relacionarlo con su entorno.

2.1.2.3 Conceptos Sobre Gráficos en 3D

Para la generación de un objeto en 3D se tiene 4 fases principales, las cuales fueron usadas para la generación del modelo virtual utilizado en este proyecto:

1. *Modelado*, el cual consiste en ir dando forma a objetos individuales que luego serán usados en la escena. Existen diversos tipos de geometría para modelador con NURBS⁶ y modelado poligonal.
2. *Iluminación*, que hace referencia a la creación de luces de diversos tipos puntuales, direccionales en área o volumen, con distinto color o propiedades. Esto genera un objeto tridimensional más apegado a la realidad al proveerle de una simulada profundidad y permite su diferenciación entre los diferentes objetos de una escena.
3. *Animación*, en donde los objetos se pueden mover en cuanto a: transformaciones básicas en los tres ejes (XYZ), rotación, escala o traslación; mediante esqueletos, dotándole al objeto de una estructura central con la capacidad de afectar la forma y movimientos de ese objeto; mediante deformadores, ya sean cajas de deformación o cualquier deformador que produzca por ejemplo deformación sinusoidal; dinámicas, para simulaciones de: ropa, pelo, dinámicas rígidas de objeto.
4. *Renderizado*, el cual es el proceso final de generar la imagen 2D o animación a partir de la escena creada. Esto puede ser comparado al tomar una foto o en el caso de la animación, al filmar una escena de la vida real. Generalmente se buscan imágenes de calidad foto realista, y para

⁶ NURBS (acrónimo inglés de la expresión Non Uniform Rational B-splines) es un modelo matemático muy utilizado en la computación gráfica para generar y representar curvas y superficies.

este fin se han desarrollado muchos métodos especiales. Las técnicas van desde las más sencillas, como: render de alambre (wireframe rendering), render basado en polígonos, hasta las técnicas más modernas como: Scanline Rendering, Raytracing, la radiosidad o el Mapeado de fotones. El proceso de renderizado necesita una gran capacidad de cálculo, pues requiere simular gran cantidad de procesos físicos complejos. La capacidad de cálculo se ha incrementado rápidamente a través de los años, permitiendo un grado superior de realismo (Foley, 1995).

2.1.2.4 Software para diseño en 3D

Varias herramientas fueron evaluadas para el presente proyecto, analizando sus características y las ventajas que generan al momento de realizar el diseño del modelo en 3D.



Figura 2.6: Google SketchUp.

- **Google SketchUp** (figura 2.6) es un programa informático de diseño y modelaje en 3D para entornos arquitectónicos, ingeniería civil, diseño industrial, GIS, videojuegos o películas.

Es un programa desarrollado y publicado por Google. Caracterizado por:

- Sencillo e Intuitivo.
- Enfocado a proyectos de varios entornos.
- Complejidad al momento de modelar cuerpos orgánicos.
- Curva de aprendizaje suave y tiempo relativamente corto.
- Tutoriales propuestos por su desarrollador.
- Dispone de herramientas básicas.

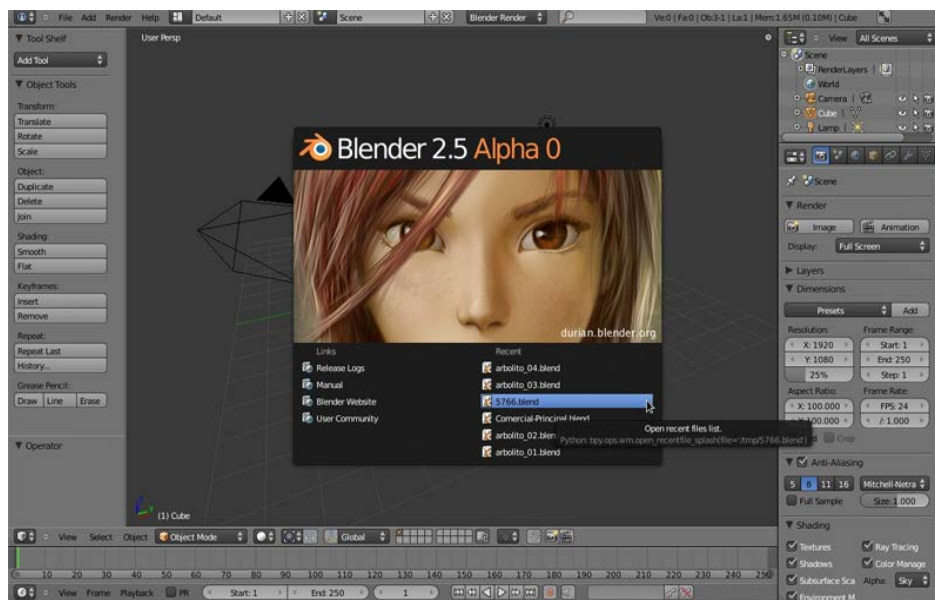


Figura 2.7: Entorno de trabajo de Blender.

- **Blender** (figura 2.7), es un programa informático multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales. El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre.

Los aspectos más sobresalientes de esta herramienta se pueden resumir en:

- Potente herramienta gratuita de modelado 3D.
- Compleja interfaz de usuario, no basada en ventanas y poco intuitiva.
- Curva de dificultad elevada

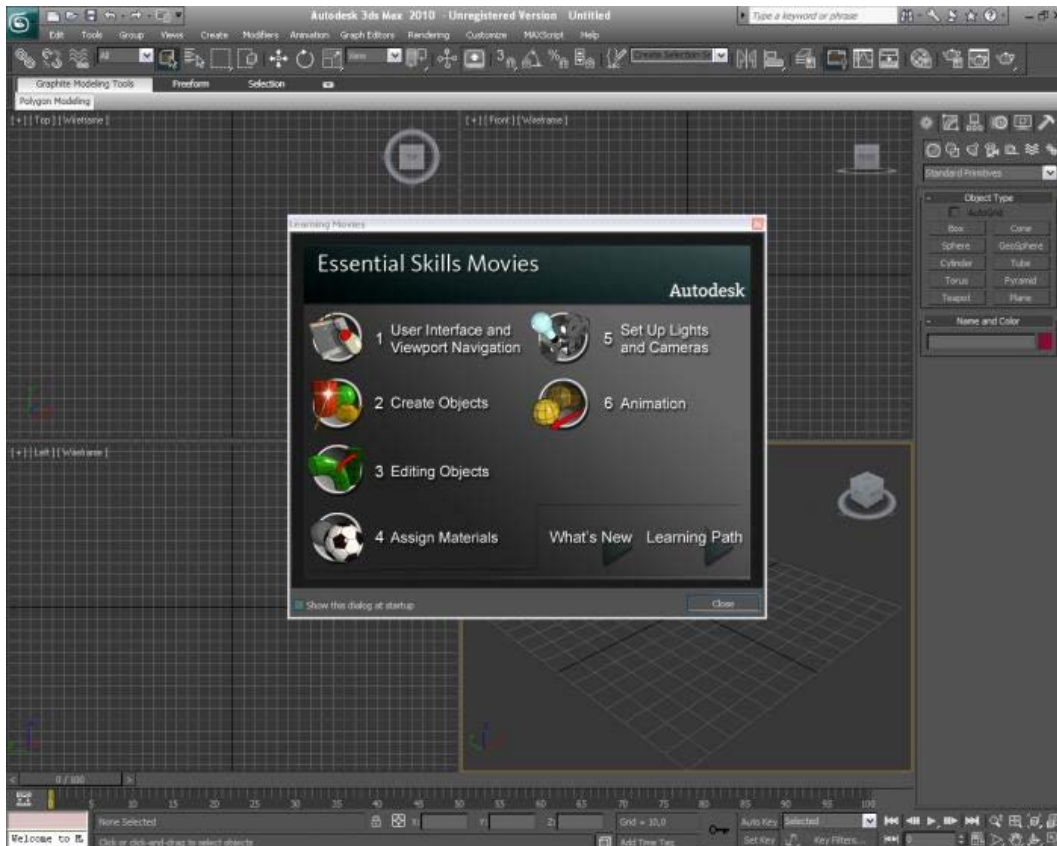


Figura 2.8: Pantalla inicial de 3ds Max 2010.

- **Autodesk 3ds Max 2010** (figura 2.8), esta es la herramienta seleccionada para el modelado de este proyecto en mención, es igualmente un software de creación de gráficos y animaciones en 3D, destacado por:
 - Arquitectura basada en complementos.
 - Diseño orientado a las edificaciones y proyectos arquitectónicos.
 - Avanzado modelamiento poligonal y mapeado de texturas.

- Elevado precio, aunque también existe la versión académica y de evaluación⁷.
- Facilidad de exportación a otros programas.

La herramienta por sí misma es secundaria en contraste con el conocimiento y la habilidad del usuario para generar un modelo en 3D, el éxito llega solo con la práctica, no se puede evadir la gran cantidad de obstáculos de la producción sin haberlos encontrado de primera mano y teniendo la tenacidad necesaria para superarlos, es una verdad inequívoca. Mientras más tiempo se le dedique al uso de la herramienta, se mejora cada vez más la habilidad con la misma y se descubren nuevos modos de realizar una acción en específica y de un modo mucho más eficaz.

2.2 METODOLOGÍA PARA EL DESARROLLO DEL APLICATIVO 3D

2.2.1 Ingeniería de Software aplicada al desarrollo de aplicativos

La Ingeniería del software es una tecnología estratificada. Se apoya sobre un enfoque de calidad. El fundamento es la capa de proceso, que se refiere a la unión que mantiene juntas las capas de tecnología que refieren el desarrollo racional y oportuno de la ingeniería del software. El proceso define un marco de trabajo para un conjunto de áreas claves de proceso que se deben establecer para la entrega efectiva de la tecnología de la ingeniería del software (Pressman, 2006).

La ingeniería de software está implícita en el desarrollo del presente proyecto, como un punto de partida para un correcto desarrollo del trabajo a realizar. El objetivo de la ingeniería de software es diseñar y no trabajar en los detalles de la

⁷ Para este proyecto se uso la versión académica.

implementación. La parte importante es que la solución pueda ser realizada y sirva para el propósito requerido (Liu, 2003).

2.2.1.1 Capas de Ingeniería de Software



Figura 2.9: Capas de Ingeniería de Software.

Tal como se indica en la figura 2.9 la ingeniería de software, debe descansar sobre un esfuerzo de organización de la calidad. La gestión total de la calidad y las filosofías similares fomentan la continua mejora de los procesos consensuados para el proyecto.

El fundamento de la ingeniería de software es la capa de procesos. El proceso define un marco de trabajo para un conjunto de áreas clave, las cuales forman la base del control de gestión de proyectos de software y establecen el contexto en el cual: se aplican los métodos técnicos, se producen resultados de trabajo, se establecen hitos, se asegura la calidad y el cambio se gestiona adecuadamente. En el presente proyecto se toma como proceso a Scrum, un marco de trabajo referencial sobre el cual se sustenta el desarrollo.

Los métodos de la ingeniería de software indican cómo construir técnicamente el software. Debido a que Scrum define un marco de trabajo ágil, enfocándose en el desarrollo y no en la documentación, en el proyecto se utiliza un método genérico de desarrollo. La figura 2.10 muestra como se abordara el análisis de requisitos, el diseño, la codificación y las pruebas de un modo continuo durante todo el desarrollo del proyecto.

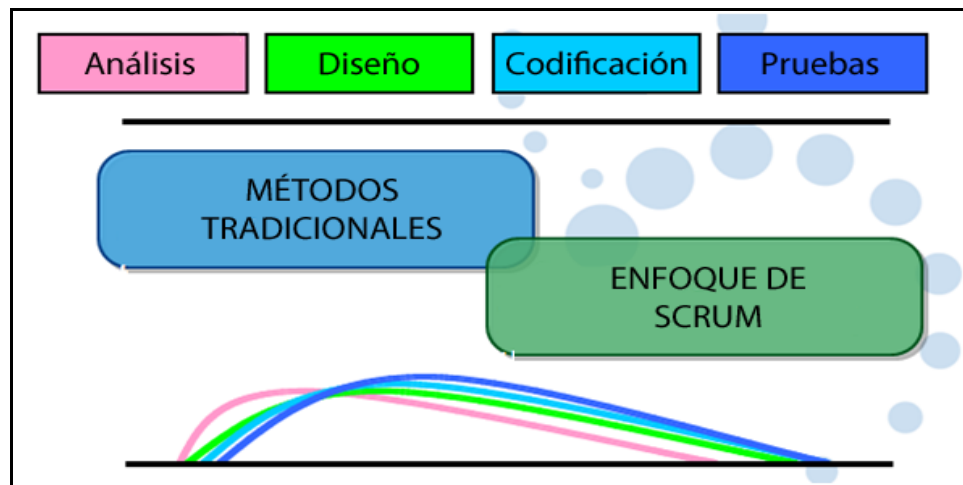


Figura 2.10: Método utilizado por Scrum

Por último, Las herramientas proporcionan un soporte para el proceso y los métodos, y en el presente proyecto será UML.

2.2.2 Lenguaje de Modelado UML

El Lenguaje Unificado de Modelado define un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y además describe la semántica esencial del significado de los diagramas y símbolos utilizados.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece los siguientes diagramas con los cuales es posible modelar sistemas.

- Diagramas de Casos de Uso: Modelado de los procesos de negocio.
- Diagramas de Secuencia: Modelado del paso de mensajes entre objetos.
- Diagramas de Colaboración: Modelado de interacciones entre objetos.
- Diagramas de Estado: Modelado del comportamiento de los objetos en el sistema.
- Diagramas de Actividad: Modelado del comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases: Modelado de la estructura estática de las clases en el sistema.
- Diagramas de Objetos: Modelado de la estructura estática de los objetos en el sistema.
- Diagramas de Interacción: Captura de la comunicación entre objetos.
- Diagramas de Componentes: Modelado de componentes del sistema.
- Diagramas de Implementación: Modelado de la distribución del sistema.

UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos.

2.2.2.1 Diagramas Recomendados

Los diagramas a representar dependerán del sistema a desarrollar, es por ello que para efecto de este proyecto se evalúan las siguientes recomendaciones, que se deberán adaptar a las características de cada desarrollo, y seguramente será la práctica lo que diga las cosas faltantes o los diagramas que parecen ser menos necesarios (Booch, 2006).

- Aplicación mono-puesto:
 - Diagrama de Casos de uso.
 - Diagrama de Clases.
 - Diagrama de Actividad.
- Aplicación mono-puesto, con entrada de eventos:
 - Diagramas de aplicación mono-puesto.
 - Añadir: Diagrama de Estados.
- Aplicación cliente servidor:
 - Diagramas de aplicación mono-puesto, con entrada de eventos.
 - Añadir: Diagrama de despliegue y diagrama de componentes.
- Aplicación compleja distribuida:
 - Todos los diagramas de UML

Para una aplicación sencilla se debe realizar entre tres y seis tipos de diagramas, y para una aplicación compleja unos nueve tipos. Para la mayoría de los casos es suficiente con tres o cuatro diagramas.

UML está diseñado para el modelado tanto de pequeños sistemas como de sistemas complejos, y se debe tener en cuenta que los sistemas complejos pueden estar compuestos por millones de líneas de código y ser realizados por equipos de centenares de programadores. Así que el presente proyecto, desde el punto de vista de UML no deja de ser un proyecto relativamente pequeño.

2.2.3 El Proceso Scrum

Scrum define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Se basa en: el desarrollo incremental de los requisitos del proyecto en bloques temporales cortos y fijos; la priorización de los requisitos por valor para el cliente y coste de desarrollo en cada iteración; la potenciación del equipo, que se compromete a entregar unos requisitos y para ello se le otorga la autoridad necesaria para organizar su trabajo; y la sistematización de la colaboración y la comunicación tanto entre el equipo y como con el cliente (Münch, 2010).

2.2.3.1 Beneficios de Scrum

El desarrollo del proyecto requiere la entrega continua de iteraciones para su evaluación y corrección de ser requerida. La Fundación Guayasamín debe evaluar el estado del proyecto a fin de realizar cambios cuando sea necesario. Scrum es beneficioso para este trabajo ya que se fundamenta en la entrega mensual de resultados y permite una gestión de las expectativas del cliente con relación al producto en desarrollo. Además Scrum permite la adaptación y flexibilidad del proyecto mediante su evaluación continua dentro de cada iteración. Scrum es flexible y adaptable y permite una mitigación de los riesgos, ya que una de las principales características de Scrum es la continua evaluación de las habilidades del equipo de trabajo y su compromiso hacia la realización de una tarea específica (Albaladejo, 2009).

2.2.3.2 El Proceso

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos⁸. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

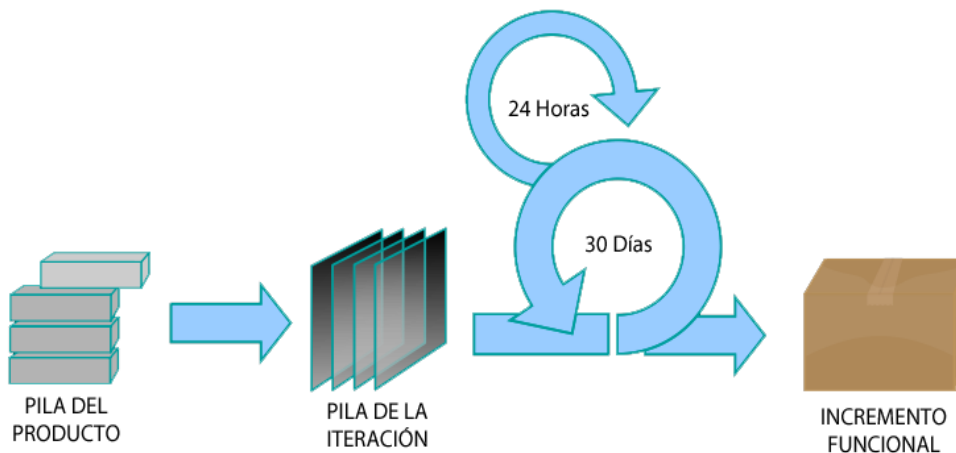


Figura 2.11: El proceso Scrum.

En la Figura 2.11 se muestra como el proceso parte de la pila del producto, que representa el catálogo completo de requerimientos del sistema a desarrollar. El cliente prioriza en esta lista los objetivos balanceando el valor que le aportan respecto a su costo y se los divide en iteraciones y entregables. De manera regular el cliente puede maximizar la utilidad de lo que se desarrolla y el retorno de inversión mediante una nueva planificación de objetivos que realiza al inicio de cada iteración. Scrum comprende varias actividades, roles y herramientas que serán analizadas a continuación.

⁸ Iteraciones de un mes aproximadamente.

2.2.3.3 Roles

Los roles y responsabilidades que Scrum define para un proyecto son los siguientes:

- **Propietario del Producto (Product Owner):** Es el representante de todas las personas interesadas en los resultados del proyecto. Se encarga de la definición de los objetivos del proyecto y colabora con el equipo para planificar, revisar y dar detalle a los objetivos de cada iteración.
- **Facilitador (Scrum Master):** Líder del equipo responsable de que todos los participantes sigan las reglas y procesos de Scrum. Asegura la disponibilidad de las herramientas en cada iteración, facilita las reuniones de Scrum y está encargado de eliminar los impedimentos que el equipo tenga para continuar con su trabajo.
- **Equipo (Team):** Es el grupo de personas que se encargan en conjunto del desarrollo del proyecto. Tienen un objetivo común y trabajan de manera equitativa para beneficio del proyecto en curso. Se considera una buena práctica de Scrum que el equipo contenga entre 5 y 9 personas (Albaladejo, 2009).
- Además, están los roles externos, que tienen cierto grado de relevancia en el proyecto Scrum, pero que no influyen directamente en el mismo:
- **Usuario (User):** Potencial usuario del sistema, se debe tomar en cuenta las características de los posibles usuarios al momento de desarrollar el aplicativo.

- **Cliente (Customer):** Grupo interesado en el desarrollo del proyecto del cual es parte el Propietario del Producto.
- **Administración (Management):** Encargados de suministrar los recursos necesarios para la realización del producto (Gregory, 2011).

2.2.3.4 Actividades

Las actividades son el listado de tareas que un equipo debe llevar a cabo dentro de una iteración, puntualmente son:

- **Reunión de Planificación del Sprint (Sprint Planning Meeting):** Donde el cliente presenta al equipo la lista de requisitos, indicando la prioridad de cada uno de ellos. Además el equipo debe realizar una planificación de cómo se realizarán y en qué momento se abordarán cada uno de los requisitos planteados, estimando tiempos y asignando tareas a cada uno de los integrantes.
- **Reunión diaria de Scrum (Daily Scrum):** A diario se debe realizar una reunión de sincronización, cada miembro inspecciona el trabajo de los demás para poder hacer las adaptaciones necesarias, actualiza el estado de la Pila de la Iteración (Sprint Backlog) y el gráfico de trabajo pendiente (Burndown charts).
- **Reunión de Revisión de la Iteración (Sprint Review Meeting):** Reunión informal donde el equipo presenta al cliente los requisitos completados en la iteración, de modo que se pueda apreciar un esbozo del producto final. Aquí el cliente puede replantear los objetivos y prioridades para la siguiente iteración o para algún cambio dentro del proyecto global.

- **Replanificación del proyecto:** Mientras se está desarrollando una iteración, el cliente puede empezar a realizar cambios a los requisitos priorizados a fin de enfocar de mejor manera la siguiente iteración del proyecto, modificándolos de la manera que crea conveniente y cambiando incluso el contexto del proyecto si así lo requiere.

2.2.3.5 Artefactos

- **Pila del Producto (Product Backlog):** Representa la visión y expectativas del cliente respecto a los objetivos y entregas del proyecto. El cliente es el responsable de crear y gestionar esta lista, y plasma en ella las expectativas de los resultados a obtener. Contiene los requisitos de alto nivel, el esfuerzo estimado en horas de trabajo para cada requisito, las posibles iteraciones y entregas a realizar, y si es necesario los riesgos que se consideren dentro de cada requisito.
- **Pila de la Iteración (Sprint Backlog):** Elaborada en la Reunión de Planificación del Sprint (Sprint Planning Meeting), es un plan para completar los requisitos seleccionados para la iteración y que se compromete a demostrar al cliente al finalizar la iteración, en forma de un entregable. Permite identificar las tareas de riesgo y los problemas que se presenten a lo largo de una iteración.

- **Gráficos de trabajo pendiente (Burndown charts):** Muestra la velocidad a la que se están completando los requisitos. Permite deducir si el tiempo es el requerido para la finalización del proyecto o iteración. Se lo puede representar en intervalos de días u horas pendientes para la culminación de las tareas de la iteración.

2.2.4 Framework XNA 3.1

XNA es un conjunto de herramientas, desarrolladas por Microsoft, que proporcionan una API para el desarrollo de videojuegos en varias plataformas⁹. Técnicamente se lo puede considerar como un marco de trabajo, basado en DirectX y .NET. Por lo tanto, permite simplificar y hacer más intuitivo el uso de las librerías nativas de DirectX y simplifica de manera notable la programación permitiendo la concentración del esfuerzo en el contenido a desarrollar.

Esta no es la única alternativa que existe para el desarrollo de un aplicativo en 3D, debido al creciente interés por el desarrollo independiente de aplicativos multimedia; existen otras compañías que también han tomado una postura similar a la de Microsoft. Se tiene como ejemplo a Apple que promueve la utilización de su SDK para sus plataformas iPhone e iPod Touch. Y no tan distante a la plataforma Windows también existen otras herramientas como: DarkBasic, Torque3D, Blitz3D entre otras, las cuales también fueron consideradas para el presente proyecto, pero la decisión de usar XNA 3.1 se la puede sustentar en:

⁹ Plataformas de Microsoft, las cuales son XBOX 360, Zune y Windows.

- Se trata de un kit de desarrollo completamente gratuito, a diferencia de las otras alternativas mencionadas para Windows. Existe sin embargo el costo implícito cuando el IDE a utilizar es Microsoft Visual Studio 2008. Para el presente proyecto se usó una licencia académica de Visual Studio.
- Se dispone de todas las herramientas de hardware necesarias, puesto que al utilizar otra plataforma como las propietarias de Apple, se incurre en la compra de los equipos y además se requiere que el usuario final también disponga de estos equipos para poder utilizar el aplicativo.
- Además existe una gran comunidad virtual con ejemplos prácticos y tutoriales enfocados en el desarrollo de aplicativos 3D con XNA.

2.2.4.1 Aspectos Generales sobre XNA

Para utilizar XNA Framework es necesario un entorno de desarrollo que, en el caso de este proyecto concreto, es Visual Studio 2008. Se trata de un entorno de desarrollo integrado (IDE) al cual se le agregó posteriormente las bibliotecas de XNA para el desarrollo del aplicativo y para poder trabajar con XNA fue necesario también el uso del lenguaje de programación C#.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

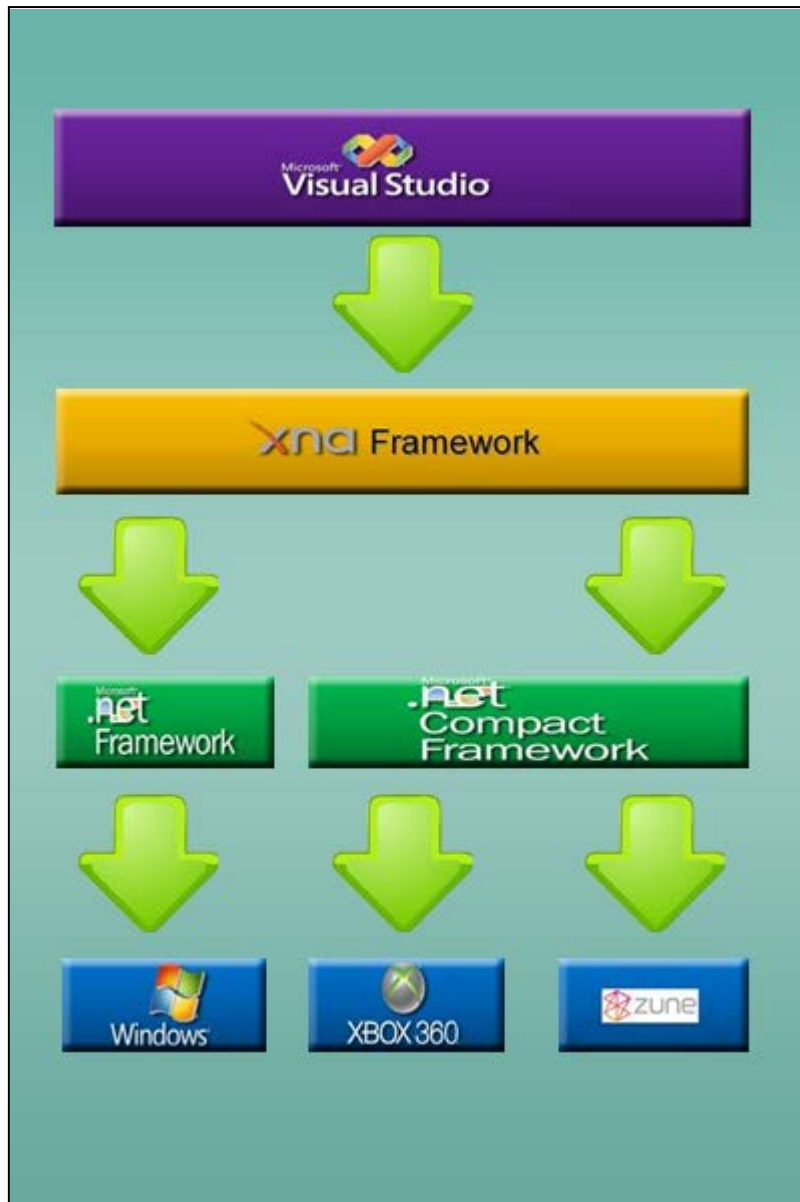


Figura 2.12: Capas de XNA.

En la figura 2.12 se muestra una estructura de las diferentes capas de XNA. Comenzando por la parte superior, como visual Studio utiliza la funcionalidad del XNA Framework, y éste a su vez se basa en el .NET Framework. Y por último están las plataformas a las que puede ser destinada la aplicación.

CAPÍTULO 3

3. ANÁLISIS, DISEÑO Y DESARROLLO

3.1 CARACTERÍSTICAS DE SCRUM

Scrum es un proceso que contiene un conjunto de prácticas y roles predefinidos, los cuales fueron debidamente determinados para el presente proyecto y se los detalla a continuación.

3.1.1 Roles

Los roles que actúan se los puede dividir en dos grupos, diferenciados por su nivel de compromiso y por el grado de involucramiento con el proyecto en desarrollo.

3.1.1.1 Roles Principales

En el primer grupo tenemos los roles que están comprometidos enteramente con el desarrollo y el éxito del aplicativo:

- ***PRODUCT OWNER (PROPIETARIO DEL PRODUCTO)***

Responsable: Sra. Verenice Guayasamín

La Sra. Verenice Guayasamín representa a la propietaria del producto, como representante de la Fundación Guayasamín. Representa a todos los interesados con el proyecto, verificando continuamente el avance del mismo, evaluando los diferentes logros alcanzados al final de cada iteración y tomando decisiones respecto a las futuras iteraciones, de tal modo que el Recorrido Virtual del Museo sea acorde a las expectativas de los Stakeholders.

- **SCRUM MASTER (FACILITADOR)**

Responsable: Ing. Cesar Villacís

Responsable de la correcta dirección de los avances del proyecto. Durante cada iteración se compromete a verificar la disponibilidad de todos los elementos que requiera el equipo para un correcto desarrollo del proyecto. Además actúa como un intermediario para facilitar una correcta comunicación entre el equipo de trabajo y la propietaria del producto.

- **TEAM (EQUIPO)**

Responsable: Luis Cóndor

Encargado del desarrollo del aplicativo en cada una de las iteraciones. Comprometido con el correcto funcionamiento del producto entregado al final de cada iteración. En concordancia con las buenas prácticas de Scrum, no está a cargo una tarea específica dentro del proyecto, realiza el modelado, codificación, evaluación y distribución del producto.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.1.1.2 Roles Secundarios

Además existen los roles externos, que tienen cierto grado de relevancia en el proyecto Scrum, y que deben ser tomados en cuenta para un mayor grado de satisfacción con el Recorrido Virtual que va a ser entregado.

- ***USER (USUARIO)***

Se entiende como usuarios a las personas ó entidades que van a recibir el CD Multimedia con el Recorrido Virtual. Dentro del desarrollo del proyecto, se enfoca el diseño de tal modo que sea fácil de usar por cualquier persona que no tenga muchos conocimientos de informática, haciendo el recorrido lo más intuitivo posible y con explicaciones claras de las características y opciones que posee el aplicativo.

- ***CUSTOMER (CLIENTE)***

El cliente representa una generalización mayor del Propietario del Producto. Dentro del presente proyecto esta personificado por la Fundación Guayasamín, entidad a la cual será entregado el aplicativo completo para su posterior distribución.

- ***MANAGEMENT (ADMINISTRACIÓN)***

En este proyecto se consideró a la administración del Museo: La Capilla del Hombre. Entidad que facilitó los elementos necesarios para el desarrollo del modelo virtual del museo, con información, fotografías y facilidades de acceso al museo.

3.1.2 Artefactos de Scrum

Scrum dictamina la utilización de 3 artefactos esenciales en un proyecto: La pila del proyecto, con los requisitos totales del aplicativo a desarrollar; la pila de la iteración, que determina los requisitos a desarrollar en cada iteración; y la gráfica de trabajo pendiente, que representa el trabaja que falta por hacer en un proyecto en el tiempo. En la tabla 3.1 se muestra la pila del proyecto, como un punto de inicio para el desarrollo del proyecto.

Tabla 3.1: Pila del producto.

ITERACIÓN	ÍTEM	DESCRIPCIÓN	ESFUERZO ESTIMADO (horas)	PRIORIDAD
1	1	Tomar las medidas físicas del museo.	5	ALTA
	2	Fotografiar las obras en exposición.	5	ALTA
	3	Modelar el museo en 3ds Max.	100	ALTA
	4	Determinar la iluminación a usar en el modelo.	4	MEDIA
	5	Realizar un video del modelo del museo.	6	BAJA
2	6	Configurar las herramientas de XNA Game Studio 3.1.	5	ALTA
	7	Desarrollar el recorrido virtual por el museo.	110	ALTA
	8	Generar el instalador del aplicativo.	3	MEDIA
	9	Documentar los manuales de usuario.	2	BAJA
3	10	Documentar la información de las obras y exposiciones.	10	ALTA
	11	Desarrollar los diferentes estados del aplicativo.	20	ALTA
	12	Integrar la base documental en el aplicativo.	30	MEDIA
	13	Compilar toda la información en un medio óptico.	10	MEDIA

Los otros dos artefactos (pila de iteración y gráfica de trabajo pendiente) se detallará individualmente dentro de cada iteración.

3.1.3 Iteraciones de Scrum

Mediante una reunión entre los tres principales roles del proyecto (Propietaria del Producto, Facilitador y Equipo) se han determinado tres iteraciones para el desarrollo del presente proyecto. La primera iteración consiste en el modelado del Museo utilizando el software 3ds Max, la segunda radica en el desarrollo del aplicativo utilizando XNA 3.1, y la tercera consta de la integración del contenido multimedia a la aplicación desarrollada.

En las siguientes secciones se procederá a detallar el trabajo realizado en cada una de las iteraciones definidas.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.2 ITERACIÓN I: DISEÑO DEL MODELO EN 3D

3.2.1 Resumen

La primera iteración del proyecto desarrollado consistió en el modelado en 3D del Museo: La Capilla del Hombre. Se trabajó en conjunto con la Propietaria del producto (Sra. Verenice Guayasamín) quien facilitó el proceso de inspección del museo y permitió la toma de fotografías y medidas de la edificación, con el objetivo de obtener un modelo muy apegado a la realidad. Con todas las mediciones y los datos obtenidos se procedió con la creación de un plano completo de las áreas a ser modeladas, y posteriormente se utilizó el software 3ds Max Studio 2010 para el modelado en 3D de la edificación, que luego sería exportado a otros medios. Con el software mencionado se renderizó un video que viene a constituir el Incremento obtenido de esta iteración.

3.2.2 Reunión de Planificación del Sprint (Sprint Planning Meeting)

Durante la primera reunión del proyecto se revisó la pila del producto que fue determinada con la ayuda de la Sra. Verenice Guayasamín, representante de la Fundación Guayasamín para el presente proyecto. Se definió como meta el desarrollo de un aplicativo desktop que permita realizar un recorrido virtual por el museo: La Capilla del Hombre de Oswaldo Guayasamín. Este desarrollo estaría distribuido en tres iteraciones, cada una con un entregable que permita el seguimiento del proyecto y determine si se puede o no continuar con la siguiente iteración.

Para esta iteración se definió el objetivo de crear el modelo del museo en 3ds Max 2010. El modelo deberá contar con: las 2 salas de exposiciones principales, una a nivel del piso y la otra en el subsuelo; el exterior del museo, que permita ver y recorrer la edificación por fuera; y un lobby intermedio que conecta a la sala del primer piso con la sala del subsuelo, aquí se puede observar el panorama de la ciudad y además el acceso externo al museo.

La iteración tomaría un esfuerzo estimado de 120 horas, el cual fue dividido a lo largo de 20 días no consecutivos, asumiendo un trabajo diario de 6 horas sin tomar en cuenta descansos ni recesos.

Se estimó que se requerirían 5 horas para la medición física del museo, 5 horas para fotografiar las obras en exposición, 100 horas para el modelado del museo en 3ds Max, 4 horas para lo correspondiente a la iluminación detallada del modelo y 6 horas para la realización de un video que recorre todas las áreas modeladas dentro de 3ds Max. Al final de esta reunión se generó la pila de la iteración.

3.2.3 Pila de la Iteración (Sprint Backlog)

La primera pila de la iteración consta de todos los requerimientos que estén relacionados con el modelado del museo en 3D. En la tabla 3.2 se detalla la pila de la iteración para los primeros 10 días, los siguientes días se detallan en la tabla 3.3.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

Tabla 3.2: Pila de la Iteración I - Primera parte.

Iteración I: Diseño del Modelo en 3D											
ÍTEM	Tarea	días de la iteración/esfuerzo restante									
		0	1	2	3	4	5	6	7	8	9
		120	115	110	104	98	97	96	92	89	78
1	Tomar las medidas físicas del museo.										
	Realizar una visita general al museo.	1	0	0	0	0	0	0	0	0	0
	Generar un plano referencial de la edificación.	2	0	0	0	0	0	0	0	0	0
	Medir la estructura del museo.	2	0	0	0	0	0	0	0	0	0
2	Fotografiar las obras en exposición.										
	Identificar las obras a fotografiar y su localización.	1	1	0	0	0	0	0	0	0	0
	Tomar fotografías de cada obra.	2	2	0	0	0	0	0	0	0	0
	Importar las fotografías y recortar las obras.	1	1	0	0	0	0	0	0	0	0
	Renombrar a los archivos para facilitar su utilización.	1	1	0	0	0	0	0	0	0	0
3	Modelar el museo en 3ds Max.										
	Modelar el primer piso.	30	30	30	24	20	19	18	16	13	2
	Modelar el piso inferior.	20	20	20	20	18	18	18	18	18	18
	Modelar el lobby intermedio.	20	20	20	20	20	20	20	18	18	18
	Modelar el exterior.	10	10	10	10	10	10	10	10	10	10
	Modelar las esculturas.	4	4	4	4	4	4	4	4	4	4
	Modelar las pinturas.	4	4	4	4	4	4	4	4	4	4
	Modelar detalles y objetos adicionales dentro del museo.	12	12	12	12	12	12	12	12	12	12
4	Determinar la iluminación a usar en el modelo.										
	Determinar el tipo de luz a usar.	2	2	2	2	2	2	2	2	2	2
	Definir los puntos donde se colocaran luces.	2	2	2	2	2	2	2	2	2	2
5	Realizar un video del modelo del museo.										
	Determinar las cámaras a usar	2	2	2	2	2	2	2	2	2	2
	Generar los puntos por los cuales se moverá la cámara	3	3	3	3	3	3	3	3	3	3
	Generar y comprimir el video.	1	1	1	1	1	1	1	1	1	1

Tabla 3.3: Pila de la Iteración I - Segunda parte.

Iteración I: Diseño del Modelo en 3D											
ÍTEM	Tarea	días de la iteración/esfuerzo restante									
		10	11	12	13	14	15	16	17	18	19
		72	62	54	36	30	22	18	10	6	0
1	Tomar las medidas físicas del museo.										
	Realizar una visita general al museo.	0	0	0	0	0	0	0	0	0	0
	Generar un plano referencial de la edificación.	0	0	0	0	0	0	0	0	0	0
	Medir la estructura del museo.	0	0	0	0	0	0	0	0	0	0
2	Fotografiar las obras en exposición.										
	Identificar las obras a fotografiar y su localización.	0	0	0	0	0	0	0	0	0	0
	Tomar fotografías de cada obra.	0	0	0	0	0	0	0	0	0	0
	Importar las fotografías y recortar las obras.	0	0	0	0	0	0	0	0	0	0
	Renombrar a los archivos para facilitar su utilización.	0	0	0	0	0	0	0	0	0	0
3	Modelar el museo en 3ds Max.										
	Modelar el primer piso.	0	0	0	0	0	0	0	0	0	0
	Modelar el piso inferior.	14	4	0	0	0	0	0	0	0	0
	Modelar el lobby intermedio.	18	18	14	0	0	0	0	0	0	0
	Modelar el exterior.	10	10	10	6	0	0	0	0	0	0
	Modelar las esculturas.	4	4	4	4	4	0	0	0	0	0
	Modelar las pinturas.	4	4	4	4	4	4	0	0	0	0
	Modelar detalles y objetos adicionales dentro del museo.	12	12	12	12	12	8	8	0	0	0
4	Determinar la iluminación a usar en el modelo.										
	Determinar el tipo de luz a usar.	2	2	2	2	2	2	2	2	0	0
	Definir los puntos donde se colocaran luces.	2	2	2	2	2	2	2	2	0	0
5	Realizar un video del modelo del museo.										
	Determinar las cámaras a usar	2	2	2	2	2	2	2	2	2	0
	Generar los puntos por los cuales se moverá la cámara	3	3	3	3	3	3	3	3	3	0
	Generar y comprimir el video.	1	1	1	1	1	1	1	1	1	0

3.2.4 Tareas

3.2.4.1 ÍTEM 1: Tomar las medidas físicas del museo.

- *Realizar una visita general al museo*

Se realizó una visita a las instalaciones para del museo: La Capilla del Hombre con el fin de conocer la edificación (Figura 3.1) y realizar un bosquejo del trabajo a realizar.



Figura 3.1: Fotografía de la parte exterior del museo.

- *Generar un plano referencial de la edificación.*

Se diseñó en papel un plano general de cada área del museo a ser modelada, para tomar anotar las medidas de la edificación.

- *Medir la estructura del museo.*

Tomando como apoyo el diseño realizado en la tarea anterior, se realizó la medición del museo y posteriormente las medidas junto con el diseño fueron pasadas al computador para tener una referencia más clara del modelo a realizar. El plano básico del museo se encuentra en el Anexo A.

3.2.4.2 ÍTEM 2: Fotografiar las obras en exposición.

- *Identificar las obras a fotografiar y su localización.*

Se requirió identificar las obras que existen en el museo y que deben ser transportadas al modelo en 3D, para lo cual se realizó un pequeño diagrama de la posición en la que se encuentran, a fin de no confundirlas al momento de realizar el modelo.

- *Tomar fotografías de cada obra.*

Se tomaron fotografías de cada una de las obras para luego poder pasarlas al modelo, se utilizó una cámara de 10 mega píxeles con el fin de obtener imágenes nítidas y de alta calidad.

- *Importar las fotografías y recortar las obras.*

Se realizó la edición de las fotografías (Figura 3.2) para poder recortar solo la obra y prepararlas para usarlas como texturas dentro del modelo en 3D. Fue necesario cambiar el tamaño de las imágenes a 1024x1024 píxeles debido a que el aplicativo presenta problemas si se usan imágenes cuyas medidas no sean una potencia de 2.

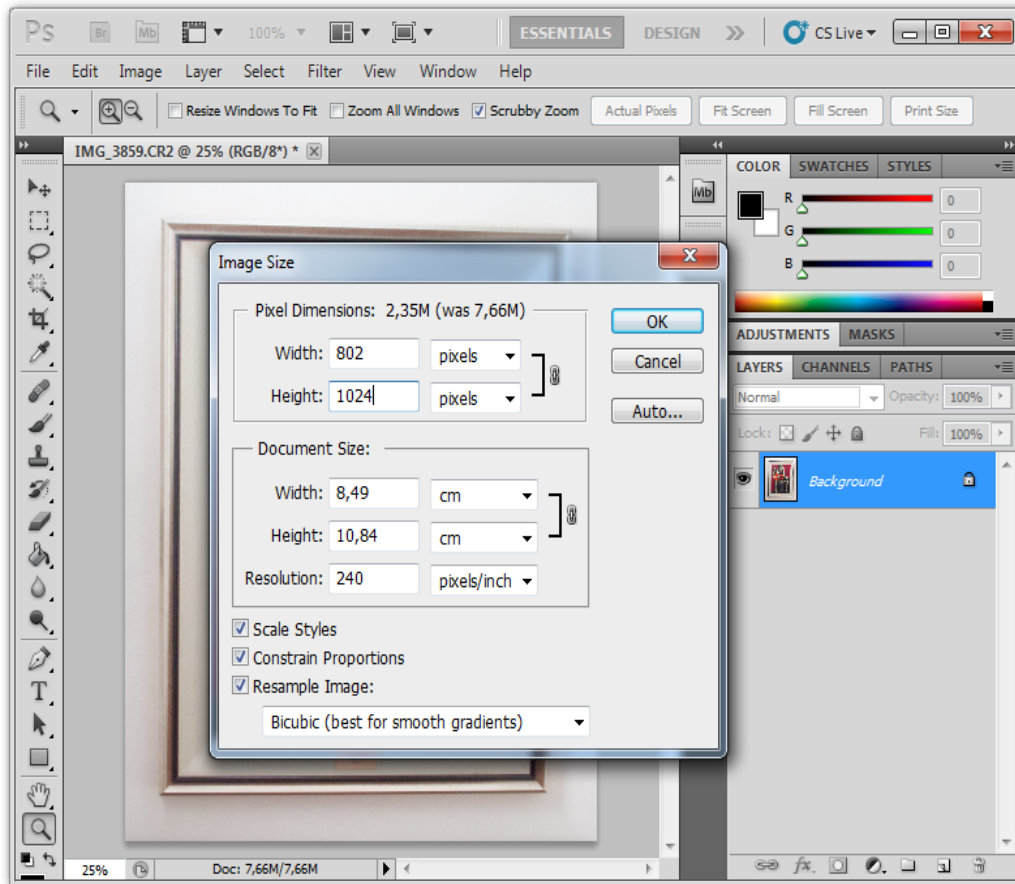


Figura 3.2: Redimensión de las imágenes en Photoshop.

- *Renombrar a los archivos para facilitar su utilización.*

Se renombró cada una de las fotos con un nombre único de archivo para evitar su confusión y poder ser ubicadas adecuadamente dentro del modelo (Figura 3.3). Es de suma importancia al momento de utilizar textura en un modelo en 3D que los nombres sean únicos y, de ser posible, separados en carpetas afines. Esto con el propósito de evitar que la herramienta cargue erróneamente una textura sobre un objeto diferente del cual fue asignado.

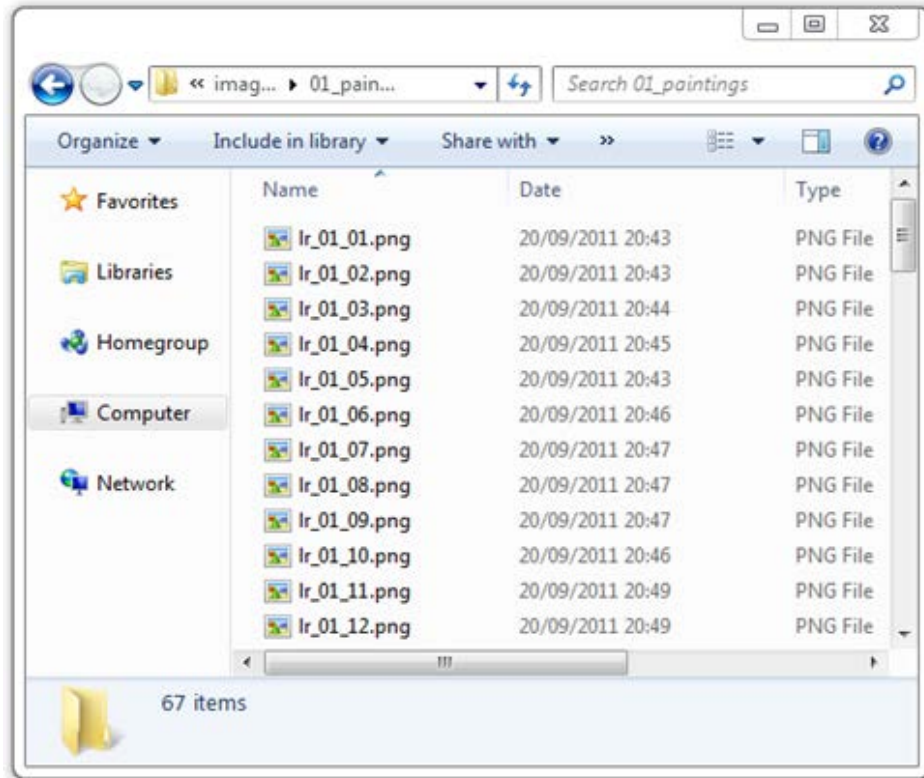


Figura 3.3: Formato de enumeración de imágenes.

3.2.4.3 ÍTEM 3: Modelar el museo en 3ds Max.

- *Modelar el primer piso.*

Se comenzó el modelado del museo por el primer piso, siendo este el menos complejo y con menor número de obras en exposición. El primer piso consta de 32 pinturas colocadas alrededor de la edificación. De éstas, 7 poseen un marco de madera especial, mientras que el resto están colocadas sobre lienzos de madera. Además incluye el área exterior del primer piso, las paredes secundarias del interior junto con las vigas que sostienen a la estructura. En la parte superior se encuentra una estructura en forma de cono, donde está colocado un mural adicional. La figura 3.4 muestra una toma del modelo terminado.



Figura 3.4: Primer piso del modelo.

- *Modelar el piso inferior.*

El modelado del piso inferior fue un reto mayor debido a la fachada que presenta la sala, pues posee varios adornos y muebles que alargan el proceso del modelado. La más compleja es una escultura llamada “silla manteña” debido al detalle que posee. Además está modelado 3 grandes murales, los que se encuentran sobre las paredes. Posee 15 pinturas principales, las cuales están esparcidas por los corredores laterales del edificio. A esto se le suma las vigas que se encuentran en el techo y el área de acceso hacia el lobby inferior. En la figura 3.5 se encuentra el modelo del piso inferior, donde se muestra la escultura “Silla Manteña” y las vigas del techo.



Figura 3.5: Piso inferior del modelo.

- *Modelar el lobby intermedio.*

El lobby intermedio posee varios muebles y pequeñas obras que fueron modeladas para que el modelo fuese fiel a su edificación real.

Lo más destacado de esta estructura es los dos grandes ventanales que existen en el lobby superior. El primero permite ver el área de ingreso, además del árbol donde se encuentran enterrados los restos del maestro Guayasamín, y el otro ventanal que permite ver el paisaje de la ciudad de Quito. Aquí se encuentra además dos esculturas del autor, y el lobby es el que permite la conexión entre el piso superior y el inferior. En la figura 3.6 se encuentra una toma del ventanal lateral desde donde se puede observar el paisaje de la ciudad de Quito.



Figura 3.6: Lobby del museo.

- *Modelar el exterior.*

El modelo exterior no supuso muchos inconvenientes ya que no posee muchos detalles que necesitasen ser modelados. Sin embargo, fue necesario actualizar el modelo e incluir el área de salía de emergencias que fue posteriormente completada. Aquí se encuentran varios murales con diferentes pinturas en gigantografías. También se puede observar el paisaje y se puede acceder directamente al área de la salida de emergencias, desde donde se puede entrar al museo y el usuario aparece en el corredor lateral del piso inferior. La figura 3.7 muestra una toma del museo desde el exterior. Se puede observar la entrada al piso superior y además las gigantografías en el lado izquierdo.



Figura 3.7: Exterior del museo.

- *Modelar las esculturas.*

Las esculturas que existen fueron un poco complejas de modelar, pues se pretendía realizar un modelo con alto detalle. De estas la más destacada es la “Silla Manteña” que se encuentra en el piso inferior. Además están las esculturas del área de la salida de emergencias y las del exterior.

- *Modelar las pinturas.*

El modelado de las pinturas fue en si el modelado de los marcos de cada obra puesto que las pinturas no eran más que texturas que se agregaban a una sección del marco modelado.

- *Modelar detalles y objetos adicionales dentro del museo.*

Como última tarea se modeló algunos objetos del museo que ayudan a darle más realismo, como son las barandas, los ascensores y los tragaluces del exterior. La figura 3.8 muestra el un área del esqueleto final del modelo.

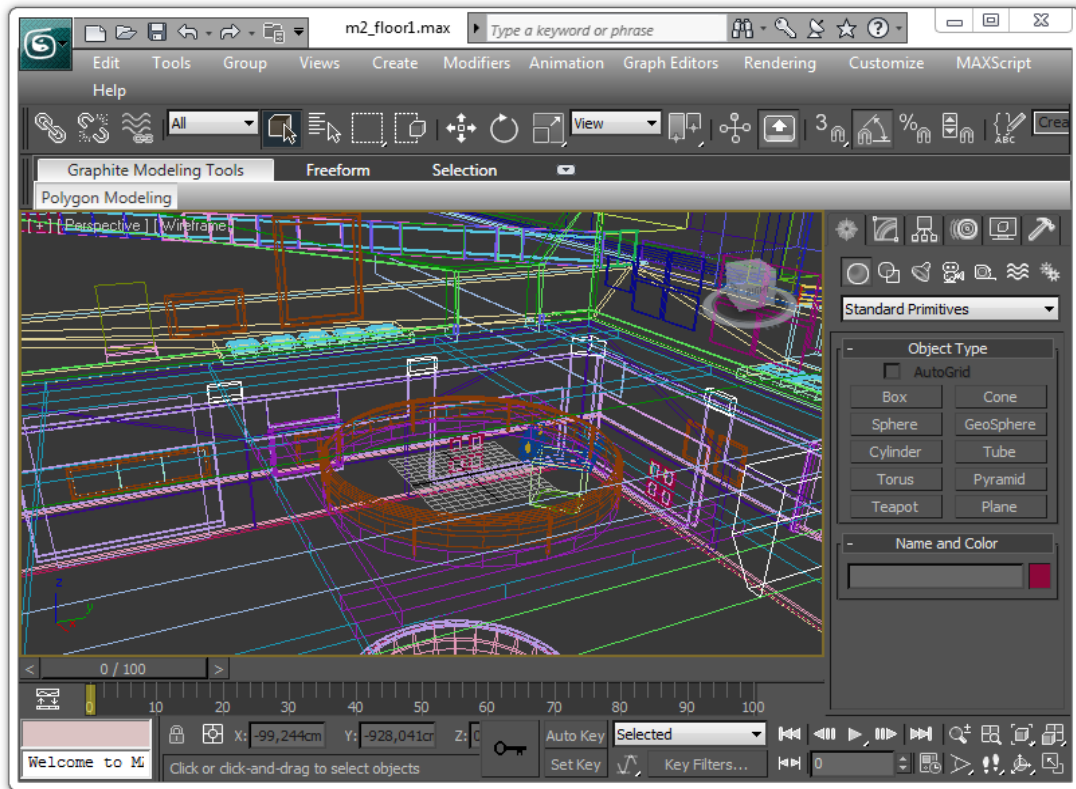


Figura 3.8: Esqueleto del museo en 3D.

3.2.4.4 ÍTEM 4: Determinar la iluminación a usar en el modelo.

- *Determinar el tipo de luz a usar.*

Una iluminación correcta del museo permitió generar un modelo más realista y sobresaltar los detalles del mismo, se utilizó el motor de renderizado: Mental Ray para la iluminación al momento de realizar el video del museo virtual. Esto permite un alto grado de realismo en las texturas y los objetos.

- *Definir los puntos donde se colocaran luces.*

Para evitar tener un modelo demasiado pesado, se utilizaron luces en puntos clave para iluminar el museo en 3D.

3.2.4.5 ÍTEM 5: Realizar un video del modelo del museo.

- *Determinar las cámaras a usar.*

La realización del video se fundamenta en el movimiento de una cámara (Figura 3.9) por dentro del modelo, por lo cual es preciso definir correctamente la cámara y sus parámetros.

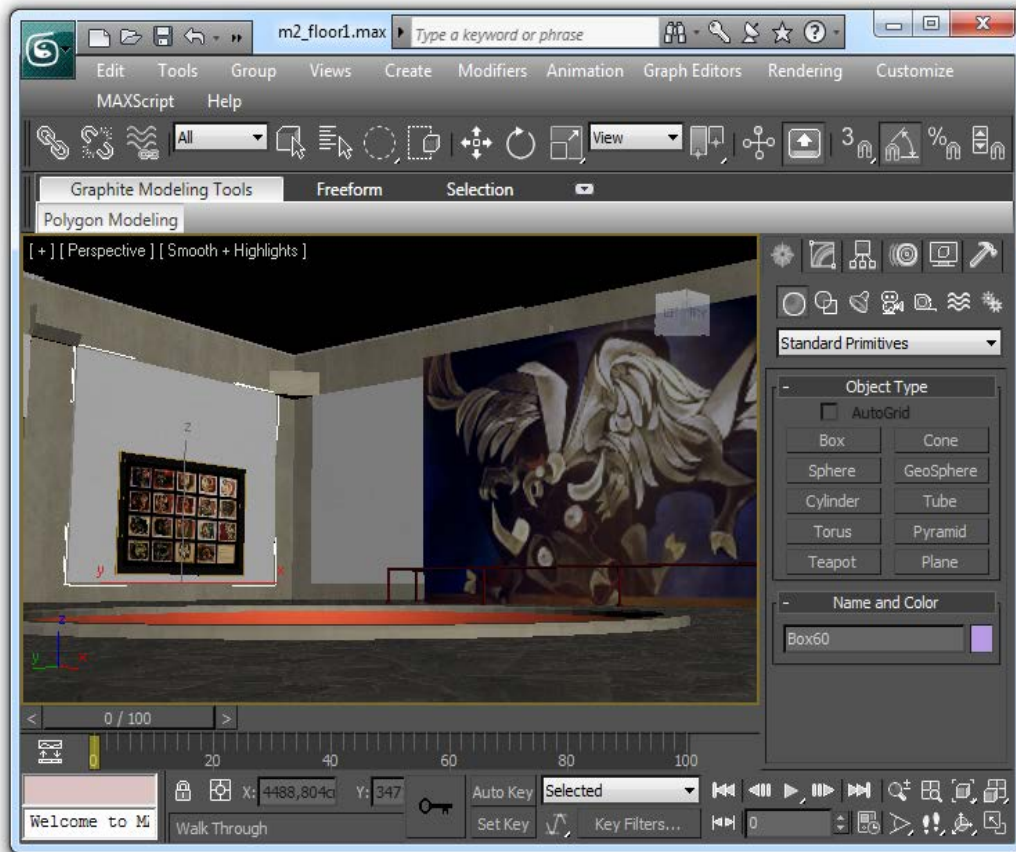


Figura 3.9: Vista del modelo con una cámara.

- *Generar los puntos por los cuales se moverá la cámara.*

Los puntos clave por los cuales se mueve la cámara deben ser definidos para que el programa genere la animación y el movimiento de la cámara a través de estos puntos referenciales.

El video realiza un giro de 360 grados de la sala superior e inferior. Luego se moviliza al lobby y al final a los corredores internos de la sala inferior.

- *Generar y comprimir el video.*

Una vez determinada la animación se la renderiza y se procede a comprimir el video a fin de que el archivo obtenido no sea demasiado grande. La animación de 50 segundos fue renderizada en aproximadamente 24 horas, por el nivel de detalle y la calidad. El video fue generado en formato WMV¹⁰.

3.2.5 Gráfica de Trabajo Pendiente

En la figura 3.6 se puede observa el esfuerzo restante desde el inicio de la iteración en el día cero, hasta el final de la iteración que fue determinado en veinte días.

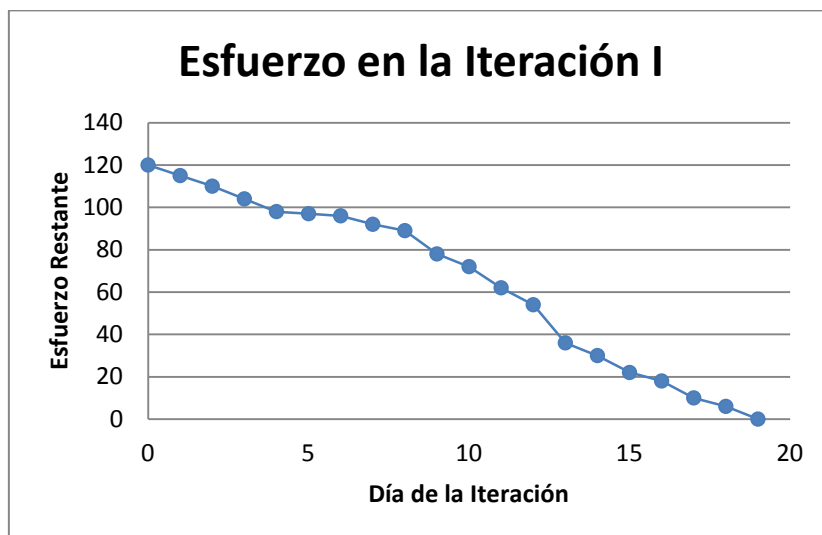


Figura 3.10: Gráfico de Trabajo Pendiente - Iteración I.

Existe un decremento constante del esfuerzo requerido en el tiempo, para lograr completar todos los requerimientos al final de los 20 días de la iteración.

¹⁰ WMV: Windows Media Video, formato de compresión de video desarrollado por Microsoft.

3.2.6 Reunión de Revisión de la Iteración (Sprint Review Meeting)

La primera reunión de revisión fue realizada para mostrar el incremento desarrollado a lo largo de la primera iteración. Junto con el Facilitador y la Propietaria del Producto se revisó el trabajo realizado. Dentro de los puntos sobresalientes de la reunión se destacan:

- Una evaluación favorable por parte de la Propietaria del Producto, quien estuvo satisfecha con el detalle alcanzado en el modelo.
- Se presentó el video generado a la Propietaria del Producto, y se le entregó para que lo pueda usar como referencia del trabajo que se está realizando
- Se vio la necesidad de utilizar menos iluminación al momento de realizar el recorrido virtual, a fin de evitar que la aplicación sea demasiado pesada, y use demasiados recursos del computador.
- Se acordó la integración del video dentro del aplicativo para que no esté aislado en un archivo por separado.
- Se acordó la próxima iteración que se realizaría una semana después, la siguiente iteración tendría como objetivo el desarrollo del aplicativo.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3 ITERACIÓN II: DESARROLLO DEL APLICATIVO SOFTWARE

3.3.1 Resumen

En esta iteración se enfoca el trabajo en la realización del software que permitirá realizar un recorrido virtual por el modelo del museo generado en la anterior iteración. Se utilizó el IDE Visual Studio 2008, junto con el Framework XNA 3.1 y el lenguaje de programación C#, estas herramientas permitieron la creación de un aplicativo de manera rápida y eficaz. En primera instancia se diseñó la cámara y el movimiento de la misma, para luego cargar las diferentes partes del museo, subdividiendo en secciones para que no exista tanto contenido cargado en memoria. Al final de la iteración se logró entregar un aplicativo que permitía recorrer por todas las salas y el exterior del museo.

3.3.2 Reunión de Planificación del Sprint (Sprint Planning Meeting)

En esta reunión se seleccionaron los requerimientos que estaban enfocados al desarrollo del aplicativo software. La Propietaria del Producto enfatizó la necesidad de que sea un recorrido fácil de manejar, debido a la diversidad de los usuarios que podrían llegar a utilizar la aplicación. La meta fue entregar un aplicativo que funcione independientemente y que utilice el modelo de la iteración anterior, junto con el correspondiente manual de usuario para que sea distribuido y pueda facilitar el uso de la aplicación.

Por segunda ocasión se estima un esfuerzo de 120 horas, dividida a lo largo de 1 mes de trabajo, sin tomar en cuenta los fines de semana.

La propietaria del producto estaría menos involucrada al inicio de esta iteración debido a que era trabajo del Equipo el poder desarrollar la aplicación, y se reuniría regularmente con la Propietaria del Producto al final, para evaluar y verificar el desempeño de la aplicación.

Se estimó 5 horas para la configuración de las herramientas de software, 110 horas para el desarrollo de la aplicación, 3 horas para generar los instaladores y 2 horas para la documentación del manual de usuario. Se generó la pila de la iteración.

3.3.3 Pila de la Iteración (Sprint Backlog)

La pila de la iteración (tabla 3.4 y tabla 3.5) está compuesta de todos los requerimientos que estén relacionados con el desarrollo del aplicativo software.

Tabla 3.4: Pila de la Iteración II - Primera parte.

Iteración II: Desarrollo del Aplicativo Software											
ÍTEM	Tarea	días de la iteración/esfuerzo restante									
		0	1	2	3	4	5	6	7	8	9
		120	115	109	102	97	91	77	73	69	63
6	Configurar las herramientas de XNA Game Studio 3.1.										
	Obtener los instaladores necesarios.	1	0	0	0	0	0	0	0	0	
	Crear una aplicación simple para probar si todo está instalado.	4	0	0	0	0	0	0	0	0	
7	Desarrollar el recorrido virtual por el museo.										
	Implementar las cámaras para la visualización de un modelo.	35	35	29	23	20	14	0	0	0	
	Cargar el modelo del museo dentro del aplicativo.	6	6	6	5	4	4	4	0	0	
	Definir el método para movimiento en el modelo.	25	25	25	25	24	24	24	24	20	
	Definir límites de movimiento dentro del modelo.	30	30	30	30	30	30	30	30	30	
	Cargar las obras dentro del modelo.	10	10	10	10	10	10	10	10	10	
	Generar el documento de diseño.	4	4	4	4	4	4	4	4	4	
8	Generar el instalador del aplicativo.										
	Definir parámetros de compilación.	1	1	1	1	1	1	1	1	1	
	Probar el aplicativo en diferentes ambientes.	2	2	2	2	2	2	2	2	2	
9	Documentar los manuales de usuario.										
	Documentar las opciones y el manejo del aplicativo.	2	2	2	2	2	2	2	2	2	

Tabla 3.5: Pila de la Iteración II - Segunda parte.

Iteración II: Desarrollo del Aplicativo Software											
ÍTEM	Tarea	días de la iteración/esfuerzo restante									
		10	11	12	13	14	15	16	17	18	19
		57	55	53	49	43	34	22	13	7	0
6	Configurar las herramientas de XNA Game Studio 3.1.										
	Obtener los instaladores necesarios.	0	0	0	0	0	0	0	0	0	0
	Crear una aplicación simple para probar si todo está instalado.	0	0	0	0	0	0	0	0	0	0
7	Desarrollar el recorrido virtual por el museo.										
	Implementar las cámaras para la visualización de un modelo	0	0	0	0	0	0	0	0	0	0
	Cargar el modelo del museo dentro del aplicativo	0	0	0	0	0	0	0	0	0	0
	Definir el método para movimiento en el modelo	8	6	4	1	0	0	0	0	0	0
	Definir límites de movimiento dentro del modelo	30	30	30	29	24	16	7	0	0	0
	Cargar las obras dentro del modelo	10	10	10	10	10	9	7	6	1	0
	Generar el documento del diseño	4	4	4	4	4	4	3	2	1	0
8	Generar el instalador del aplicativo.										
	Definir parámetros de compilación	1	1	1	1	1	1	1	1	1	0
	Probar el aplicativo en diferentes ambientes	2	2	2	2	2	2	2	2	2	0
9	Documentar los manuales de usuario.										
	Documentar las opciones y el manejo del aplicativo	2	2	2	2	2	2	2	2	2	0

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3.4 Diseño

3.3.4.1 Introducción

El recorrido virtual permite al usuario el libre movimiento por las instalaciones del museo: “La Capilla del Hombre de Guayasamín”. Implementa la posibilidad de obtener información adicional de ciertas figuras que se encuentran en exposición.

3.3.4.1.1 Lema publicitario

“Museo Virtual Interactivo: La Capilla del Hombre”

3.3.4.1.2 Controles

El usuario debe utilizar las teclas de direccionales del teclado para desplazarse por el museo, con las teclas arriba y abajo puede desplazarse para adelante y retroceder, respectivamente. Con las teclas izquierda y derecha el usuario puede girar en dicha dirección. La interfaz gráfica del usuario le indica el lugar en el cual está situado, además del objeto seleccionado con el mouse, una guía de las teclas de las opciones con las cuales puede acceder a la ayuda, volver al punto inicial ó salir del programa.

3.3.4.1.3 Personajes

El único personaje que existe dentro del recorrido virtual es el usuario, al cual no se lo puede visualizar en pantalla debido a que es un recorrido en primera persona, la cámara está situada a 1,60 metros de altura del piso, simulando la estatura promedio de una persona que realiza la visita.

3.3.4.1.4 Niveles

El recorrido se subdivide en 5 áreas: El exterior del museo, la sala principal, el lobby, el subsuelo, y la salida. Cada nivel es delimitado a fin de que no se cargue en memoria todo el modelo del museo, sino que se muestra solo el área en la cual está el usuario.

3.3.4.1.5 Acciones

El usuario puede hacer un clic izquierdo sobre algunos de los objetos dentro del museo para obtener información adicional acerca de los mismos, esta función está reservada para las obras en exposición dentro del museo.

3.3.4.1.6 Movimientos

El movimiento se lo realiza utilizando las teclas direccionales del teclado, el usuario no podrá atravesar paredes, ni los objetos grandes dentro del recorrido. El movimiento entre los diferentes niveles del museo es automático, al igual que la salida hacia la parte exterior.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3.4.2 Especificación de requisitos de software

El documento es revisado diariamente para realizar los cambios y modificaciones que se requieran a lo largo del proceso de desarrollo.

3.3.4.2.1 Descripción

La especificación de requisitos provee una lista descriptiva de las especificaciones funcionales y no funcionales de los componentes del presente software. Es derivado de la pila de la iteración (Tabla 3.4) y de las reuniones realizadas con los StakeHolders¹¹.

3.3.4.2.2 Perspectiva

Es un recorrido de una sola persona creado con el propósito de mostrar al usuario las obras y exposiciones más relevantes dentro del museo “La Capilla del Hombre de Guayasamín”. El recorrido incorpora características similares de un juego en primera persona, al permitir al usuario desplazarse por el modelo virtual e interactuar con los objetos que se encuentran dentro del mismo.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

¹¹ persona o entidad que es afectada o concernida por las actividades realizadas

3.3.4.2.3 Vista de las funciones del sistema

1. Pantalla Inicial
2. Introducción
 - a. Controles
 - b. Ayuda
3. Recorrido Virtual
 - a. Interfaz Gráfica del Usuario
 - b. Opciones
4. Video
5. Créditos
6. Base Documental

3.3.4.2.4 Características del usuario

El recorrido está diseñado para usuarios de cualquier edad. Sin embargo, dentro del recorrido se analizan las diferentes obras del maestro Oswaldo Guayasamín, por lo cual el usuario debe, en lo posible, conocer la historia y el tipo de obra que realiza el autor.

3.3.4.2.5 Restricciones

El presente recorrido no corresponde a una completa digitalización de todas las obras creadas por el maestro Oswaldo Guayasamín, sino que es un punto de partida dentro de la trayectoria del autor.

3.3.4.2.6 Suposiciones y dependencias

El ejecutable del programa se incluye dentro del CD Multimedia, y se requiere la instalación del Framework XNA 3.1 para su ejecución, el cual está incluido dentro del CD.

3.3.4.2.7 Requisito mínimo

El recorrido funciona en una computadora personal con una resolución mínima de 800x600, la pantalla debe soportar una resolución (profundidad de color) de 32-bits como mínimo. Se requiere una PC equipada con un procesador Pentium IV o superior, y 1Gb de memoria RAM, la computadora debe poseer mouse y teclado para navegar e interactuar dentro del aplicativo.

3.3.4.2.8 Interfaz del usuario

El recorrido cuenta con mensajes de ayuda para que el usuario sepa la función de las diferentes teclas. Además existen botones en pantalla para navegar por el menú principal. La aplicación se ejecuta en una ventana de 800x600 pixeles de tamaño.

3.3.4.2.9 Interfaz de hardware

El mouse permite la selección de las pinturas y la navegación por el menú, mientras que el teclado permite el movimiento y el acceso a las opciones de ayuda, salida o regreso.

3.3.4.2.10 Interfaz de software

Framework XNA 3.1

3.3.4.2.11 Interfaz de comunicación

Se requiere de un teclado y un mouse para interactuar con la aplicación.

3.3.4.2.12 Requisitos funcionales

Tabla 3.6: Requisitos funcionales del sistema.

Requisito	Descripción
1	El usuario debe poder moverse utilizando las teclas direccionales.
2	El usuario no podrá atravesar las paredes dentro del recorrido.
3	El usuario no podrá atravesar el piso.
4	Se le permitirá al usuario moverse libremente por el exterior y el interior del museo.
5	El usuario podrá desplazarse por los diferentes niveles del museo (inferior y superior).
6	El programa deberá realizar el movimiento vertical al momento de subir o bajar un nivel.
7	El usuario podrá caminar por la salida de emergencia hacia el sitio arqueológico.
8	El usuario podrá regresar al punto inicial a cualquier momento.
9	El usuario podrá obtener ayuda sobre las funciones y los controles que posee el recorrido.
10	El usuario podrá salir a cualquier momento del recorrido.
11	El usuario podrá dar clic sobre las obras en exposición para acceder a los datos de la base documental.
12	El usuario podrá recorrer las diferentes obras similares a la seleccionada dentro de la base documental.
13	El usuario no podrá mover la orientación vertical de la cámara, a fin de facilitar el movimiento y evitar confusiones.
14	El usuario se desplazara a una velocidad lenta para facilitar el movimiento dentro del museo.

3.3.4.2.13 Disponibilidad

El recorrido del museo estará disponible al final de la segunda iteración, y al terminar la tercera iteración los complementos multimedia también estarán disponibles. El aplicativo no requiere de instalación y se podrá ejecutar de inmediato desde el CD. Además se lo puede descargar desde la página web: www.MuseoVirtualCDH.com

3.3.4.2.14 Mantenibilidad

Para realizar alguna modificación se debe realizar el cambio dentro del modelo en 3D, además de volver a compilar y generar el ejecutable.

3.3.4.3 Casos de Uso

Documentan el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Para la aplicación en mención se tiene el caso de uso del recorrido virtual, el mismo que se indica en el gráfico 3.11.

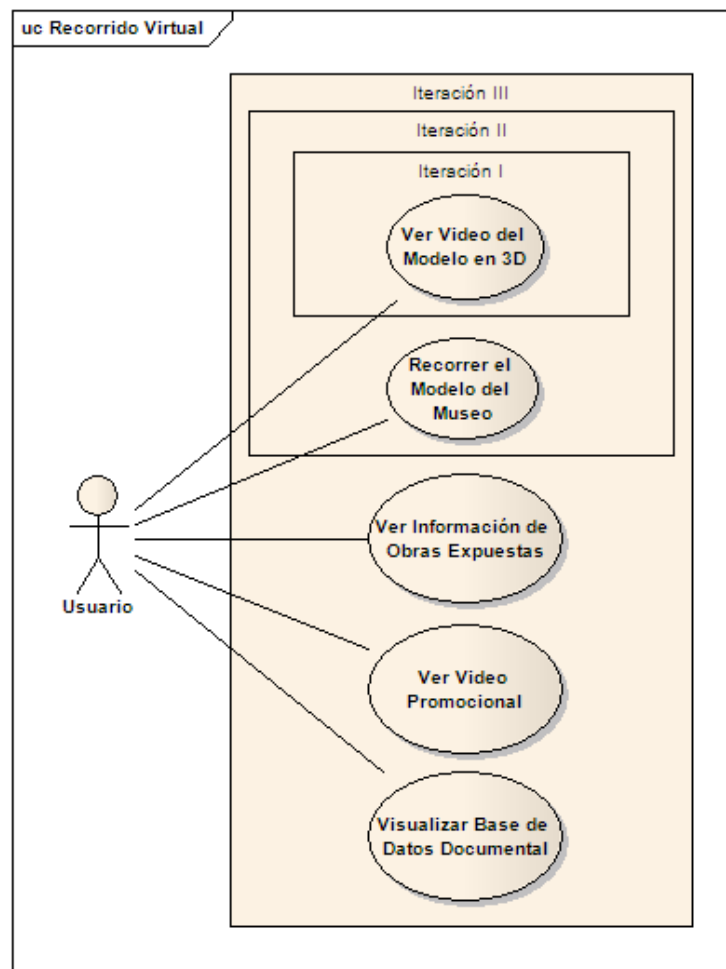


Figura 3.11: Casos de Uso.

Caso de uso 1

- *Nombre del caso de uso:*

Ver video del modelo en 3D.

- *Actor(es):*

Usuario.

- *Precondición(es):*

El usuario debe abrir el video del museo.

- *Desencadenante(s):*

El usuario selecciona el video.

- *Curso principal:*

1. El video del recorrido se muestra al usuario fuera de la aplicación principal.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3.4.3.1 Caso de uso 2

- *Nombre del caso de uso:*

Recorrer el modelo del museo.

- *Actor(es):*

Usuario.

- *Precondición(es):*

Debe ejecutar el aplicativo y estar en el menú principal.

- *Desencadenante(s):*

El usuario selecciona la opción de realizar el recorrido.

- *Curso principal:*

1. El usuario visualiza la pantalla de introducción y la ayuda del recorrido.
2. El usuario se desplaza por el museo virtual.
3. El usuario sale el recorrido presionando la tecla "S".

- *Curso Alternativo:*

1. El usuario puede volver al inicio del recorrido.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3.4.3.2 Caso de uso 3

- *Nombre del caso de uso:*

Ver información de obras expuestas.

- *Actor(es):*

Usuario.

- *Precondición(es):*

Debe estar dentro del recorrido virtual.

- *Desencadenante(s):*

El usuario da clic sobre una obra en exposición.

- *Curso principal:*

1. Se muestra en pantalla la obra seleccionada.
2. El usuario utiliza los controles para visualizar la información de la obra.
3. El usuario selecciona la opción de salir y volver al recorrido.

- *Curso Alternativo:*

1. El usuario puede seleccionar las obras similares a la seleccionada.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3.4.3.3 Caso de uso 4

- *Nombre del caso de uso:*

Ver video promocional.

- *Actor(es):*

Usuario.

- *Precondición(es):*

Debe ejecutar el aplicativo y estar en el menú principal.

- *Desencadenante(s):*

El usuario selecciona la opción de ver el video.

- *Curso principal:*

1. Se muestra en pantalla los controles de reproducción del video.
2. El usuario da clic en el botón de reproducción del video.
3. El usuario selecciona la opción de volver al menú principal.

- *Curso Alternativo:*

1. El usuario pausa o reinicia la reproducción del video.
2. El usuario regresa al menú principal.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3.4.3.4 Caso de uso 5

- *Nombre del caso de uso:*

Visualizar base de datos documental.

- *Actor(es):*

Usuario.

- *Precondición(es):*

Debe ejecutar el aplicativo y estar en el menú principal.

- *Desencadenante(s):*

El usuario selecciona la opción de la base documental.

- *Curso principal:*

1. Se muestra en pantalla los controles de navegación de la base documental.
2. El usuario selecciona la obra a visualizar.
3. El usuario selecciona las obras similares a mostrar.
4. El usuario se desplaza por la información textual de la obra.
5. El usuario regresa al menú principal.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3.4.4 Diagramas

3.3.4.4.1 Diagrama de clases

- **DIAGRAMA GENERAL**

La figura 3.12 indica las *clases del aplicativo Recorrido Virtual*

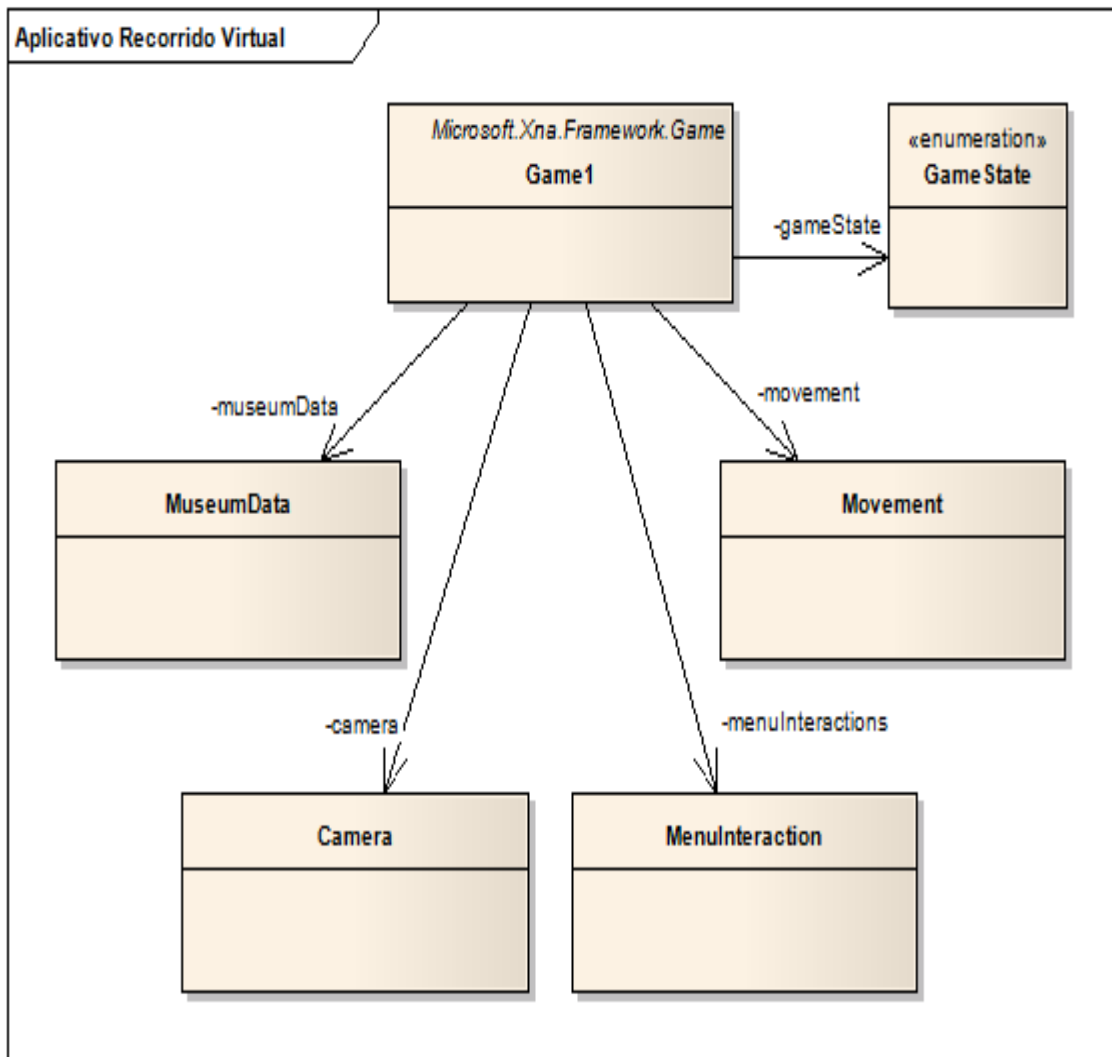


Figura 3.12: Clases de Aplicativo.

En la Figura 3.13 se indica los métodos y atributos de la clase Game1.

- **Clase “Game1”**



Figura 3.13: Métodos y atributos de la clase Game1.

En la figura 3.14 se indica los Métodos y atributos de la clase Camera

- **Clase "Camera"**

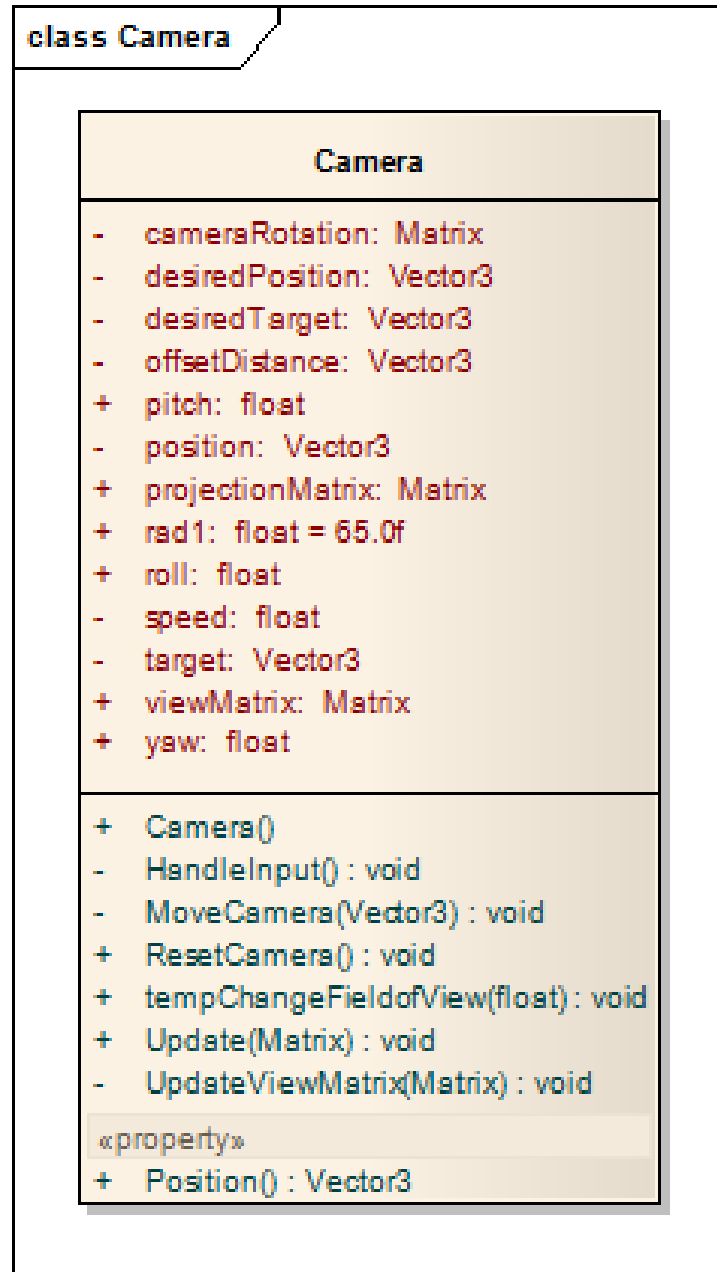


Figura 3.14: Métodos y atributos de la clase Camera.

En la Figura 3.15 se indica los Métodos de la clase MenuInteraction

- **Clase “MenuInteraction”**

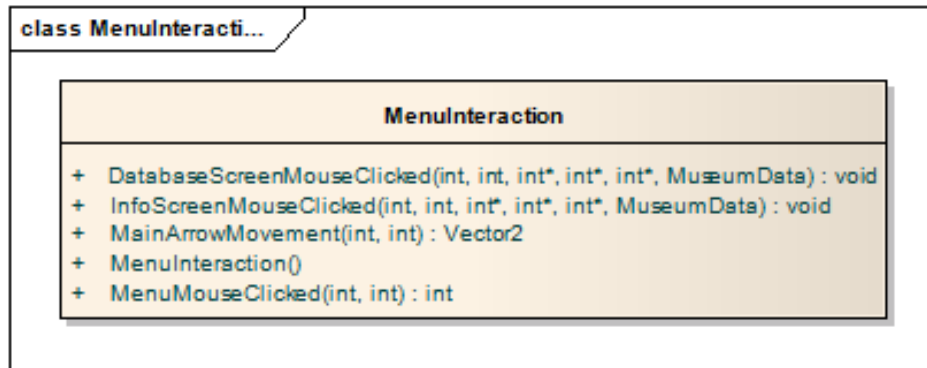


Figura 3.15: Métodos de la clase MenuInteraction.

En la Figura 3.16 se indica los métodos y atributos de la clase MuseumData

- **CLASE “MuseumData”**

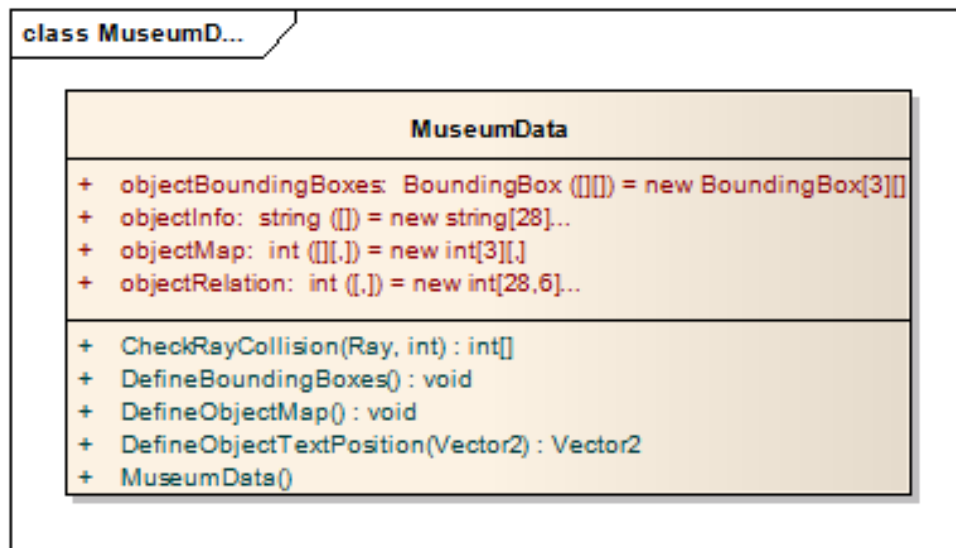


Figura 3.16: Métodos y atributos de la clase MuseumData.

En la Figura 3.17 se indica los métodos y atributos de la clase Movement

- **Clase “Movement”**

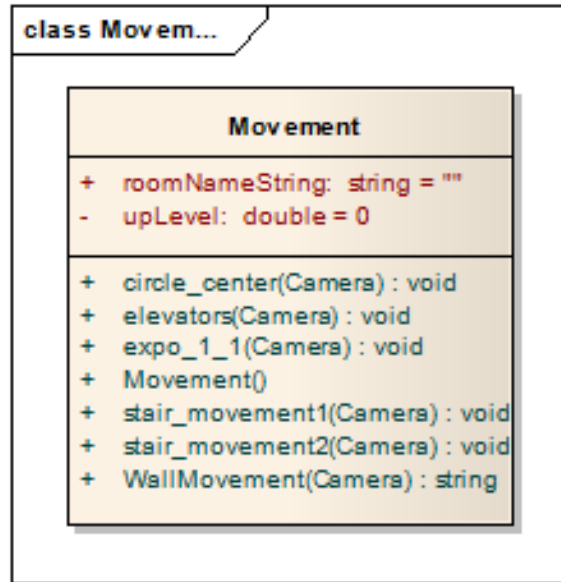


Figura 3.17: Métodos y atributos de la clase Movement.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3.4.4.2 Diccionario de clases

- **Clase Camera**

Tabla 3.7: Métodos y atributos de la clase Camera.

Clase Camera	
Descripción:	
Clase que controla la cámara, sus atributos y el movimiento de la misma.	
Atributos:	
-float radFieldOfView	Angulo de visión en radianes.
-Vector3 position	Posición de la cámara en un plano tridimensional.
-Vector3 desiredPosition	Posicion de la cámara luego del movimiento.
-Vector3 target	Objetivo al cual apunta la cámara.
-Vector3 desiredTarget	Objetivo de la cámara luego del movimiento.
-Vector3 offsetDistance	Distancia entre la cámara y el objetivo.
+float yaw, pitch, roll	Variables usadas para la rotación de la cámara.
-float speed	Velocidad con la que se mueve la cámara.
-Matrix cameraRotation	Matriz de rotación de la cámara.
+Matrix viewMatrix, projectionMatrix	Matriz de vista y proyección de la cámara.
Métodos:	
+Camera()	Constructor de la clase.
-void HandleInput()	Método para manejar el movimiento en base a las entradas por teclado.
-void MoveCamera(Vector3 addedVector)	Realizar el movimiento de la cámara.
+void ResetCamera()	Reiniciar la cámara a la posición inicial.
+void tempChangeFieldofView(float variation)	Cambio temporal del ángulo de visión.
+void Update(Matrix chasedObjectsWorld)	Actualiza la matriz de la cámara.
-void UpdateViewMatrix(Matrix chasedObjectsWorld)	Actualiza la matriz de la vista de la cámara.

- **Clase Game1**

Tabla 3.8: Atributos de la clase Game1.

Clase Game1 – Deriva de Microsoft.Xna.Framework.Game	
Descripción:	
Clase principal derivada de la clase Microsoft.Xna.Framework.Game, posee todas las propiedades y habilidades de cualquier juego de XNA.	
Atributos:	
-GraphicsDeviceManager graphics	Controla la configuración y la administración del dispositivo gráfico.
-SpriteBatch spriteBatch	Permite dibujar un grupo de sprites.
-Camera camera	Instanciación de la clase Camera.
-Movement movement	Instanciación de la clase Movement.
+Matrix[] museumMatrix	Arreglo de matrices referenciales de cada modelo.
+Model[] museumModel	Arreglo de matrices, Contiene al modelo del museo.
-SpriteFont spriteFontRoom	Textura de fuente utilizada en el nombre de la sala.
-SpriteFont spriteFontObject	Textura de fuente utilizada en el nombre de la obra seleccionada con el mouse.
-string locationString	Nombre de la sala en la cual se encuentra la cámara.
-Texture2D[] screenImage	Arreglo de Texturas bidimensionales, que contienen las imágenes de cada una de las ventanas.
-Texture2D arrowImage	Imagen de una flecha usada en el menú principal.
-Texture2D borderImage	Imagen del borde de la ventana.
-Vector2 arrowPosition	Vector bidimensional que indica la posición de la flecha usada en el menú principal.
-int infoAvailable = -1	Indica si el mouse está en una obra con información.
-int menuSelection = 0	Indica la opción seleccionada en el menú principal.
-bool imagesLoaded = false	Indica si se cargaron las imágenes dentro de la base documental, para evitar que se vuelvan a cargar.
-GameState gameState;	Estados en los que se puede encontrar la aplicación, muestra diferentes ventanas dependiendo del estado.
-KeyboardState keyBoardState	Estado del teclado al inicio de un ciclo.
-KeyboardState previousKeyBoardState	Estado del teclado en el ciclo anterior.
-MouseState mouseState	Estado del mouse al inicio del ciclo.
-MouseState previousMouseState	Estado del mouse en el ciclo anterior.
-MenuInteraction menuInteractions	Instanciación de la clase MenuInteraction.
-MuseumData museumData	Instanciación de la clase MuseumData.
-int objectIndex	Índice referencial de un objeto.
-int imageIndex	Índice de la imagen de un objeto.
-int textIndex	Índice del texto de un objeto.
-int roomIndex	Índice del lugar de localización de un objeto.
-Texture2D imageDisplayed	Define la imagen de un objeto que se dibujará.
-Texture2D textDisplayed	Define el texto del objeto que se está dibujando.
-Texture2D videoTexturePromo	Textura donde se cargara el video del museo.
-Video videoPromo	Variable donde se asigna el video del museo.
-VideoPlayer videoPlayerPromo	Variable usada para reproducir el video del museo.

Tabla 3.9: Métodos de la clase Game1.

Clase Game1 – Deriva de Microsoft.Xna.Framework.Game	
Constructores y Métodos:	
+Game1()	Constructor de la clase Game1.
#void Initialize()	Realiza la inicialización de varios atributos antes de correr la aplicación.
#void LoadContent()	Carga todo el contenido que se usara en la aplicación.
#void UnloadContent()	Realiza la descarga del contenido cuando se requiera.
#void Update(GameTime gameTime)	Permite ejecutar la lógica de la aplicación.
#void Draw(GameTime gameTime)	Usado para dibujar en pantalla la aplicación.
-int[] ObjectSelectionWithMouse(int mouseX, int mouseY)	Indica si el mouse se encuentra sobre una obra que posee información adicional.
-Ray GetSelectionRay(int mouseX, int mouseY)	Utilizado para trazar un rayo y detectar si el mouse se encuentra apuntando a un objeto.
-void LoadImagesDatabase()	Carga la imagen y el texto de un objeto en la ventana de información o en la base documental.
-void DrawModel(Model model, Matrix worldMatrix)	Permite dibujar un modelo en pantalla.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

- **Clase MenuInteraction**

Tabla 3.10: Métodos de la clase MenuInteraction.

Clase MenuInteraction	
Descripción:	
Clase que controla la interacción del usuario en el menú principal, y en las ventanas de información y base documental.	
Métodos:	
+MenuInteraction(){}	Constructor de la clase.
+Vector2 MainArrowMovement(int mouseX, int mouseY)	Permite identificar el lugar del mouse en el menú principal y dibujar la flecha de selección en el junto a la opción marcada.
+int VideoMouseClicked(int mouseX, int mouseY)	Detecta el botón en el cual se dio clic en la ventana del video.
+int MenuMouseClicked(int mouseX, int mouseY)	Detecta la opción en la cual se hizo clic en el menú principal.
+void DatabaseScreenMouseClicked(int mouseX, int mouseY, ref int objectIndex, ref int imageIndex, ref int textIndex, MuseumData objectsInteraction)	Detecta cual botón se seleccionó con el mouse en la ventana de la base documental. Cambia el objeto, la imagen o el texto a mostrarse dependiendo de cual botón se seleccionó con el mouse.
+void InfoScreenMouseClicked(int mouseX, int mouseY, ref int objectIndex, ref int imageIndex, ref int textIndex, MuseumData objectsInteraction)	Identifica el botón que se presionó con el mouse en la ventana de información adicional, y cambia el índice del texto o de la imagen correspondiente al objeto que se está visualizando.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

- **Clase MuseumData**

Tabla 3.11: Métodos y atributos de la clase MuseumData.

Clase MuseumData	
Descripción:	
Clase que controla la relación entre los objetos del museo y la base documental.	
Atributos:	
+BoundingBox[][] objectBoundingBoxes = new BoundingBox[3][];	Arreglo que permite almacenar la información de las cajas de colisión donde se encuentran objetos que poseen información adicional que puede ser mostrada al usuario.
+int[[],] objectMap = new int[3][,];	Arreglo con la información de la posición de las cajas de colisión, el objeto al cual referencia, los límites en el espacio vectorial y el índice de una imagen específica.
+string[] objectInfo = new string[28]	Arreglo que contiene los nombres de cada objeto.
+int[.,] objectRelation = new int[28,6]	Variable que relaciona a los objetos, con su información y sus imágenes correspondientes.
Métodos:	
+MuseumData(){}	Constructor de la clase.
+void DefineObjectMap()	Define el mapa de los objetos con sus límites dentro del espacio vectorial.
+void DefineBoundingBoxes()	Crea una lista de cajas de colisión en base a la información de la variable <i>objectMap</i>
+Vector2 DefineObjectTextPosition(Vector2 objectSize)	Determina la posición correcta en la que se dibujará la variable seleccionada de <i>objectInfo</i> .
+int[] CheckRayCollision(Ray pickRay, int roomIndex)	Detecta si el mouse está apuntando a un objeto dentro del museo con información adicional, mediante el uso de un rayo que inicia en el punto del mouse y que verifica si encuentra alguna caja de colisión (de la variable <i>objectBoundingBoxes</i>) en su camino.

- **Clase Movement**

Tabla 3.12: Métodos y atributos de la clase Movement.

Clase Movement	
Descripción:	
Clase que controla los límites por los cuales se puede desplazar la cámara, evitando el traspaso por las paredes y por ciertos objetos del museo.	
Atributos:	
<code>-double upLevel = 0;</code>	Variable que controla el movimiento de subida o bajada de las escaleras del Lobby del museo.
<code>+ string roomNameString = "";</code>	Variable que almacena el nombre de la sala en la que se encuentra la cámara.
Métodos:	
<code>+Movement()</code>	Constructor de la clase.
<code>+string WallMovement(Camera camera)</code>	Función que controla el movimiento de la cámara para evitar que atraviese las paredes.
<code>+void stair_movementX(Camera camera)</code>	Método que permite el movimiento de arriba hacia abajo o viceversa en las escaleras del Lobby. Utiliza una ecuación de la recta entre dos puntos X, Y para permitir el movimiento.
<code>+void wall_X_X(Camera camera)</code>	Método que identifica a las paredes que deben ser verificadas para que la cámara no las traspase.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3.4.4.3 Diagramas UML

- **DIAGRAMAS DE ESTADO**

Representa el ciclo de vida de un objeto: los eventos que le ocurren, sus transiciones, y los estados que median entre los eventos.

En la figura 3.18, se encuentra el Diagrama del estado inicial, del menú principal

ESTADO INICIAL: Refleja el estado de la aplicación en el menú principal.

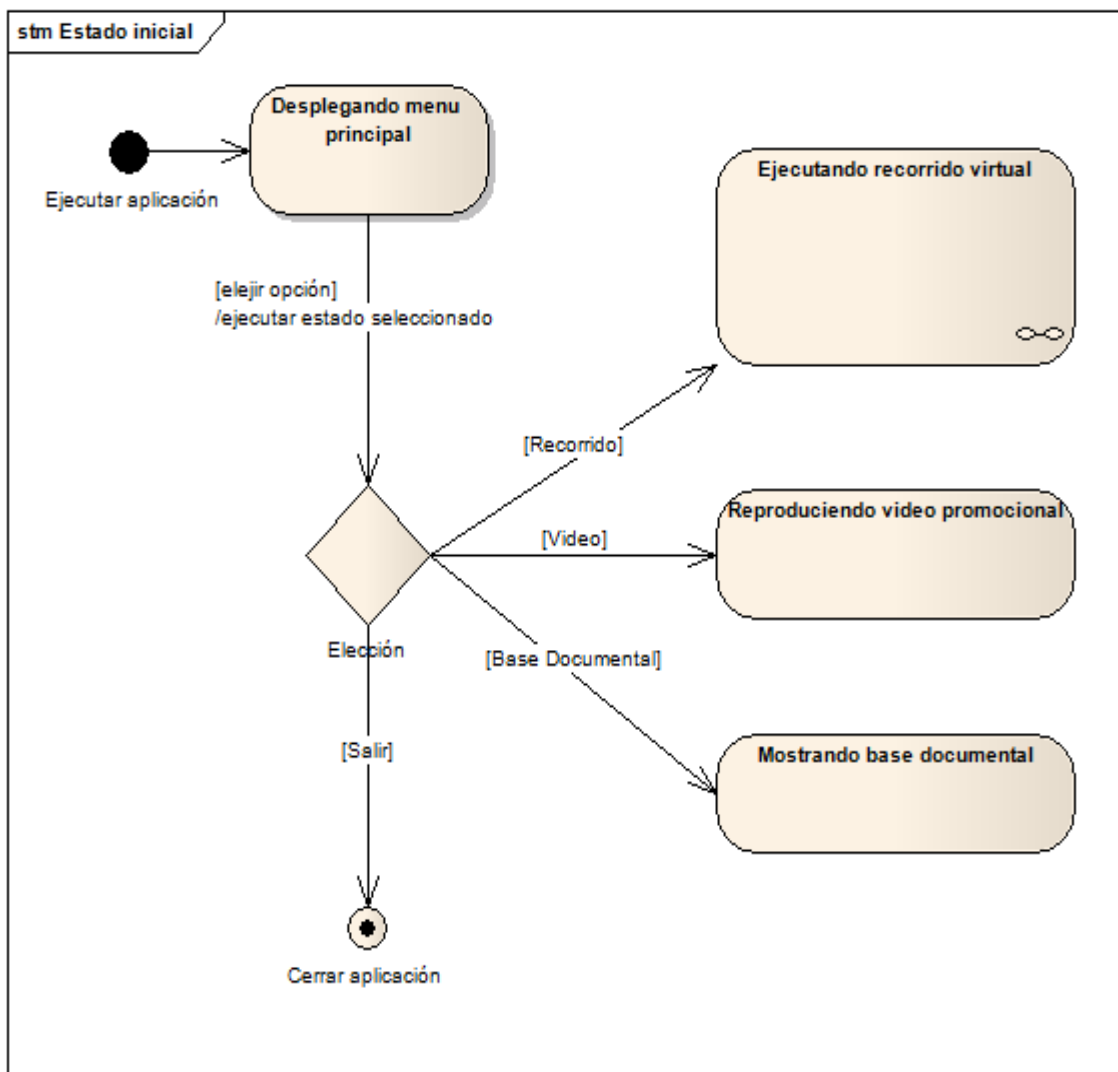


Figura 3.18: Diagrama del estado inicial.

En la Figura 3.19, se indica el *ESTADO DE LA EJECUCIÓN DEL RECORRIDO*: Refleja los diferentes estados en los cuales puede encontrarse la aplicación al momento de realizar el recorrido por el museo virtual.

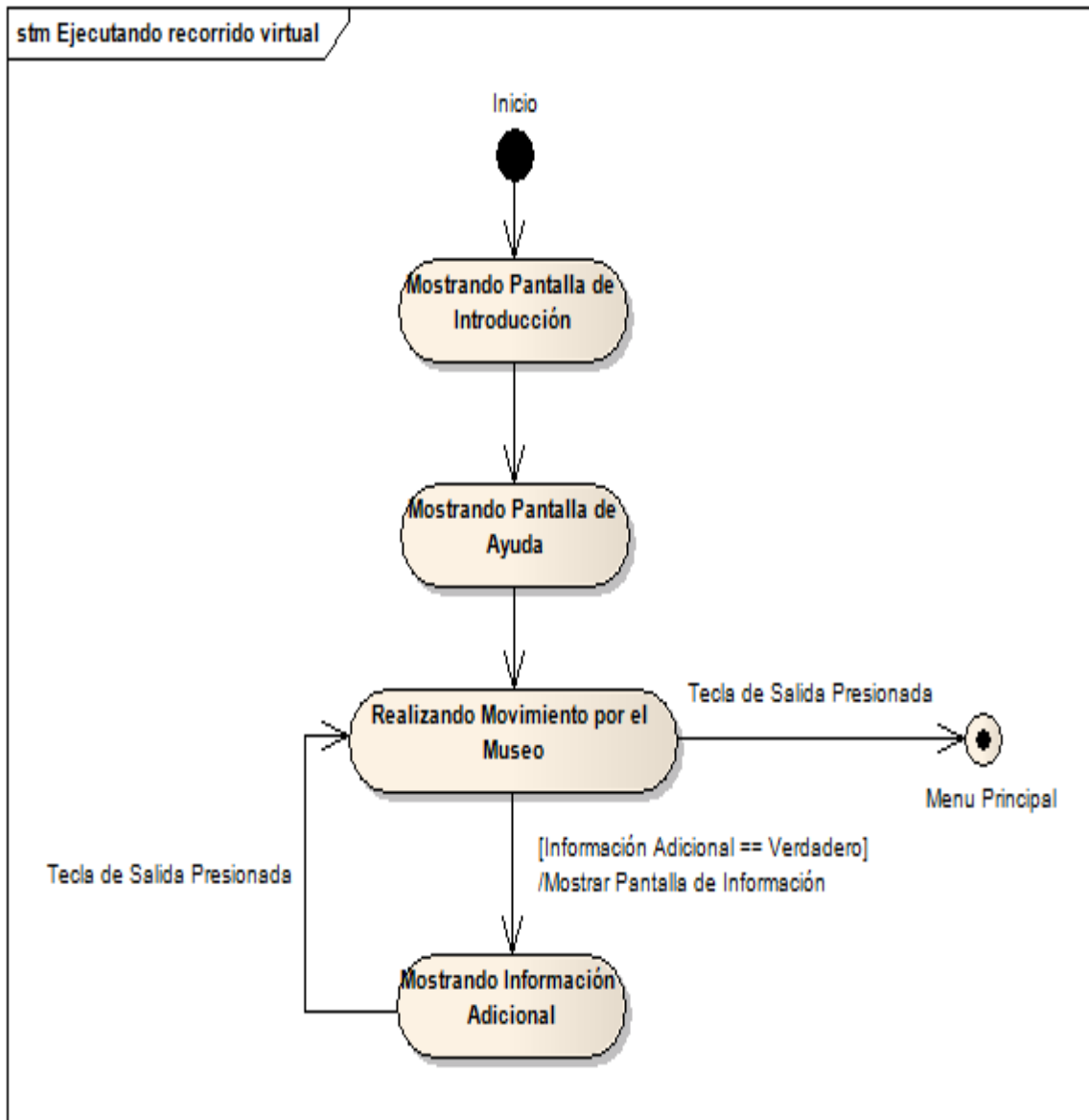


Figura 3.19: Diagrama del estado de la ejecución del recorrido.

- **DIAGRAMAS DE ACTIVIDAD**

RECORRIDO VIRTUAL: La figura 3.20 indica el diagrama que muestra las actividades que realiza la aplicación al ingresar al recorrido virtual del museo, detallando las acciones implementadas para moverse y mostrar en pantalla el modelo del museo.

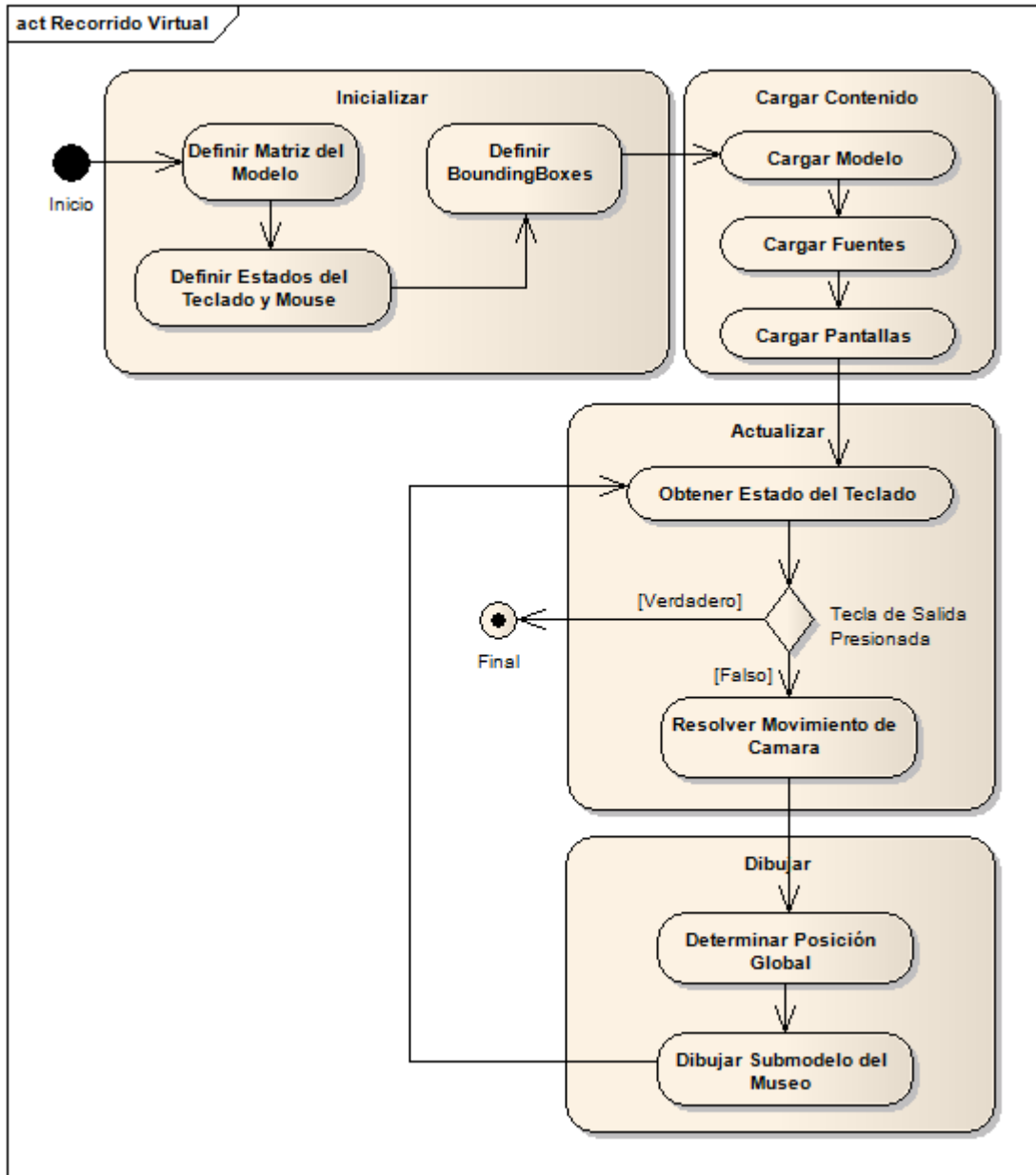


Figura 3.20: Diagrama de actividad.

3.3.5 Tareas

3.3.5.1 ÍTEM 6: Configurar las herramientas de XNA Game Studio 3.1.

- *Obtener los instaladores necesarios.*

De la página oficial de XNA se obtuvo el Game Studio 3.1, que cuenta con todo lo necesarios para utilizar XNA dentro de Visual Studio 2008 y se procedió a su instalación.

- *Crear una aplicación simple para probar si todo está instalado.*

Para verificar que todo este correcto se desarrollo una simple aplicación que mostrara un modelo en 3D en pantalla, así se pudo verificar que todo fue instalado correctamente (Figura 3.21).



Figura 3.21: Modelo en 3D de prueba.

3.3.5.2 ÍTEM 7: Desarrollar el recorrido virtual por el museo.

- *Implementar las cámaras para la visualización de un modelo.*

Uno de los aspectos fundamentales de una aplicación en 3D es la definición de las cámaras, pues sin ellas no se puede visualizar nada en pantalla, como primer punto se desarrollo la clase de la cámara que permita visualizar todo lo que se dibuje en pantalla. La clase “*Camera*” fue desarrollada durante esta tarea.

- *Cargar el modelo del museo dentro del aplicativo.*

Como segundo punto es necesario cargar el modelo dentro de la aplicación (Figura 3.22), Se realizó una división del modelo en 3ds Max para que los archivos no fuesen tan pesados.



Figura 3.22: Modelo del museo sin las obras.

- *Definir el método para movimiento en el modelo.*

El movimiento se basa en los ejemplos de XNA, que utiliza una matriz para proyectar el modelo con determinados parámetros, el movimiento fue implementado utilizando la clase “*Camera*” dentro de la clase principal “*Game1*”.

- *Definir límites de movimiento dentro del modelo.*

El movimiento debía ser controlado de tal forma que no se atravesen paredes ni los techos, se implemento varios métodos que controlan la posición de la cámara y evitan que atravesen los límites establecidos. La implementación de estas restricciones se lo hizo en la clase “*Movement*”.

- *Cargar las obras dentro del modelo.*

Una parte fundamental para que el aplicativo no consuma demasiados recursos fue la separación de los elementos del modelo, al final de la etapa de desarrollo se definió el modo por el cual se van a cargar las obras para ser presentadas en pantalla (Figura 3.23).

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE



Figura 3.23: Obras cargadas dentro del modelo.

- *Generar el documento del diseño.*

Durante el proceso de desarrollo se modifica constantemente los parámetros finales del producto y al final se obtiene un documento con las características del diseño de la aplicación.

3.3.5.3 ÍTEM 8: Generar el instalador del aplicativo.

- *Definir parámetros de compilación.*

Se configuró el IDE para que todos los archivos requeridos estén dentro del paquete de la aplicación y así no existan problemas al momento de ejecutar el aplicativo en otro computador (Figura 3.24).

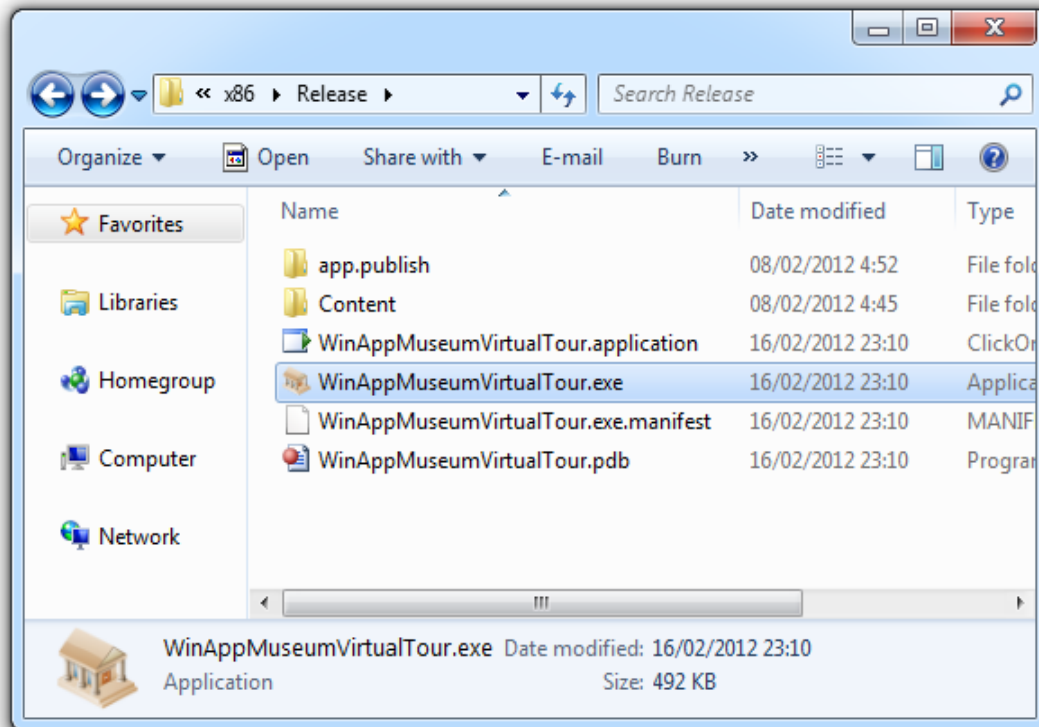


Figura 3.24: Archivos generados.

- *Probar el aplicativo en diferentes ambientes.*

Se realizaron pruebas en varias máquinas a fin de comprobar que la aplicación funcionaba sin ningún problema en diferentes versiones de Windows (XP, Vista, 7). Las pruebas permiten localizar los errores del software, con este proceso se va a verificar y revelar la calidad de la aplicación, para lo cual se ejecutarán las siguientes pruebas:

- Prueba de recuperación.
- Prueba de Resistencia.
- Prueba de Rendimiento.

Prueba de recuperación.

Para esta prueba se va a forzar al software para evaluar su recuperación. Si la recuperación es automática hay que evaluar la corrección de la inicialización, de los mecanismos de recuperación del estado del sistema, de la recuperación de los datos y del proceso de volver a iniciar la maquina y si la recuperación no es automática, hay que evaluar los tiempos medios de reparación para determinar si están dentro de unos límites aceptables.

Tabla 3.13: Pruebas de recuperación.

Tipo de Arranque	Caso de Prueba	Resultado esperado	Resultado obtenido	Observaciones
Desde la unidad de CD	Se ejecuta la aplicación y se apaga a propósito el CPU.	El sistema no debe sufrir ningún daño.	El sistema no sufrió ningún daño.	La recuperación es inmediata.

La tabla 3.13 muestra los resultados y no existe ningún problema al realizar esta prueba, debido a que la aplicación corre directamente del CD y no modifica ningún archivo de sí mismo.

Prueba de resistencia.

En esta prueba de resistencia es donde se enfrenta a la aplicación con situaciones atípicas, se va a incrementar las frecuencias de datos de entrada, ejecutar casos que requieran el máximo de memoria, realizar pruebas de sensibilidad

intentando descubrir combinaciones de datos dentro de una clase de entrada válida que pueda producir inestabilidad o un proceso incorrecto. La tabla 3.14 muestra el resultado de estas pruebas.

Tabla 3.14: Pruebas de resistencia.

Tipo de Arranque	Caso de Prueba	Resultado esperado	Tiempo respuesta	Observaciones
Desde la unidad de CD	Procesador Intel® Core TM 2 Duo, 2.4 Ghz, 2 GB memoria RAM, Windows Vista	El sistema trabaja normalmente	Rápido	Ninguna
Desde la unidad de CD	Procesador Intel® Core TM 2 Duo, 2.4 Ghz, 2 GB memoria RAM, Windows 7	El sistema trabaja normalmente	Rápido	Ninguna
Desde la unidad de CD	Procesador Intel Pentium IV, 3.0 Ghz, 1 GB en RAM, 200GB en disco, Windows XP	El sistema trabaja normalmente	Normal	Ninguna

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

Prueba de rendimiento

En esta prueba se va a medir el rendimiento del software en tiempo de ejecución. La figura 3.25 muestra el rendimiento del Equipo sin ejecutar la aplicación.

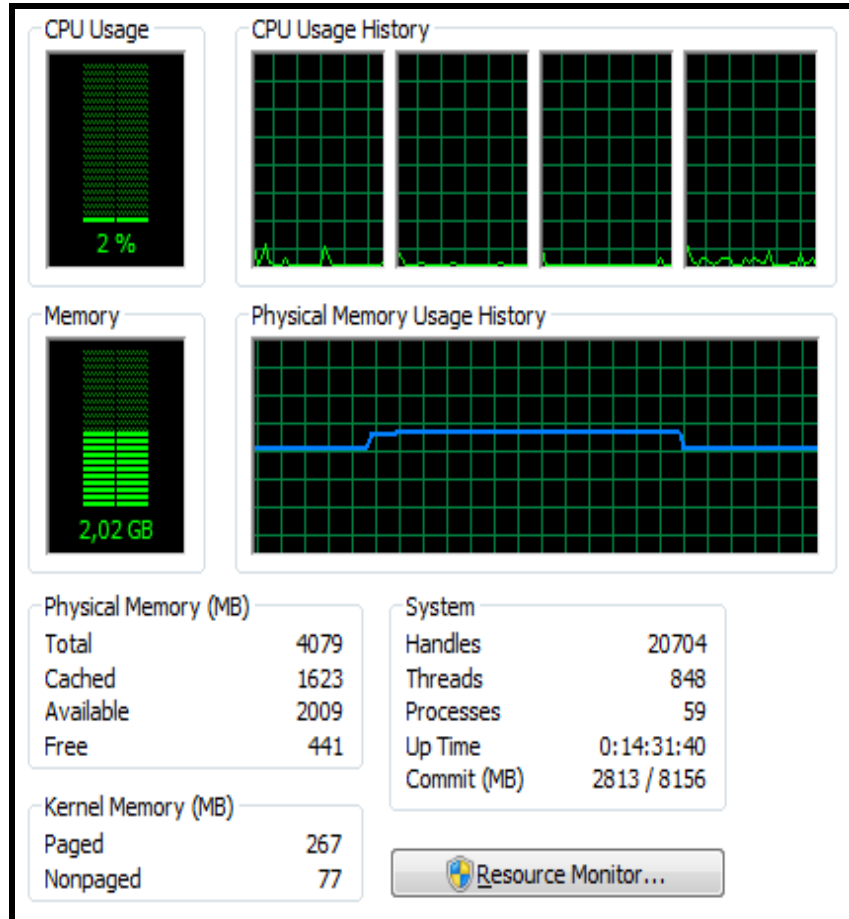


Figura 3.25: Rendimiento del equipo sin ejecutar la aplicación.

De la figura expuesta se concluye que el uso de la memoria permanece constante con picos de hasta el 2% que corresponden a la ejecución de pequeños procesos propios del sistema operativo.

Cuando se ejecuta la aplicación desde la unidad de CD y se realiza el recorrido, el rendimiento se incrementa, como se indica en la figura 3.26.

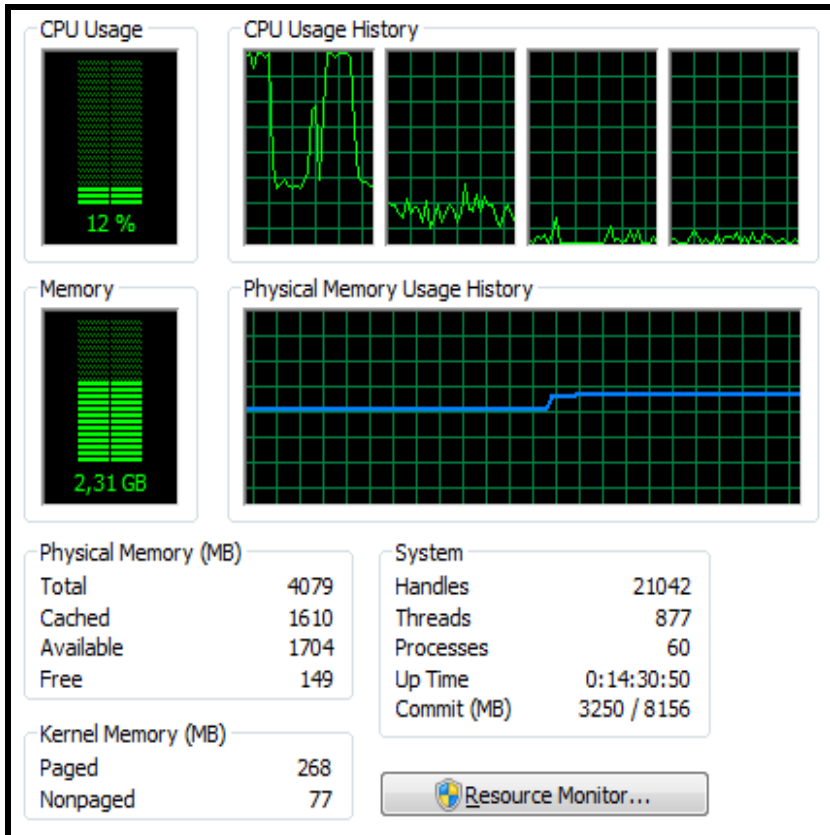


Figura 3.26: Rendimiento del equipo mientras se ejecuta la aplicación.

Como se indica en la figura 3.26 el consumo de la memoria aumenta su porcentaje y permanece constante al ejecutar la aplicación desde la unidad de CD. Se puede ver picos de 100% de uso del CPU al iniciar la aplicación los cuales luego permanecen alrededor del 30% durante la ejecución del recorrido, y al salir al menú principal estos bajan a un 12%.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.3.5.4 ÍTEM 9: Documentar los manuales de usuario.

- *Documentar las opciones y el manejo del aplicativo.*

La aplicación fue desarrollada tomando en cuenta que la interfaz debe ser lo más simple e intuitiva posible, a fin de que el usuario no tenga problemas al momento de usarla, por lo cual el manual de usuario es simple, corto y posee todas las instrucciones necesarias para manejar la aplicación. Debido a que el aplicativo fue modificado en la tercera iteración, el manual de usuario (Anexo D) refleja la última versión del mismo.

3.3.6 Documentos generados

Para un correcto desarrollo del aplicativo se desarrollaron varios diagramas UML que conceptualizan la aplicación desarrollada. Los diagramas fueron sujetos a cambios durante la etapa de desarrollo y la versión final de los mismos puede ser vista en el Anexo C.

3.3.7 Gráfica de trabajo pendiente

En la figura 3.28 se puede observar el esfuerzo restante desde el inicio de la iteración en el día cero, hasta el final de la iteración que fue determinado en veinte días.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

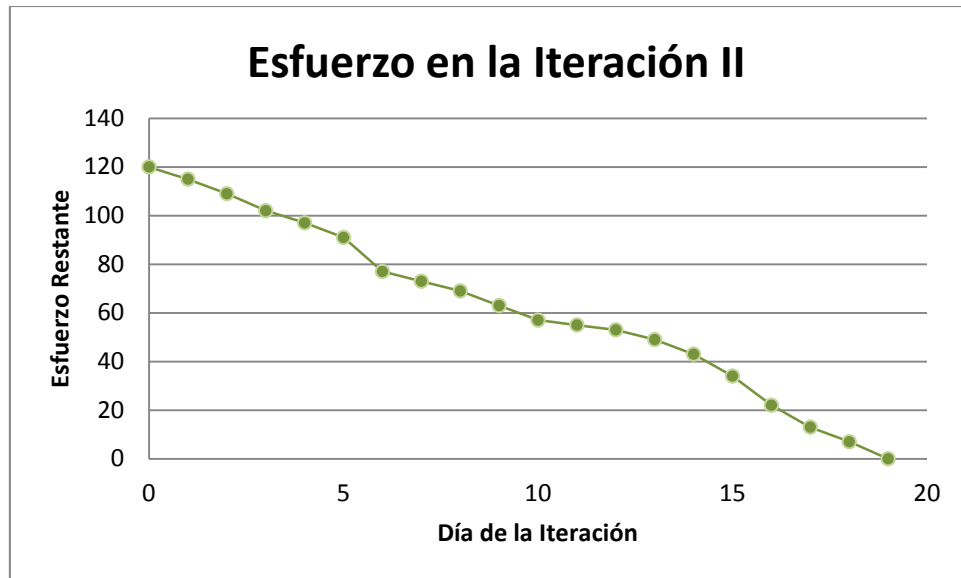


Figura 3.27: Gráfico de Trabajo Pendiente - Iteración II.

El gráfico 3.27 muestra un decremento lento de la reducción del esfuerzo, lo cual se explica debido al tiempo requerido en la documentación y las pruebas al momento de desarrollar el aplicativo. Se logró culminar exitosamente la iteración cumpliendo los requerimientos especificados.

3.3.8 Reunión de revisión de la iteración (Sprint Review Meeting)

La segunda reunión de revisión fue realizada para mostrar el incremento obtenido que en este caso es la aplicación con el recorrido virtual ya terminado, se destacan los siguientes puntos de la reunión:

- Fue favorable la evaluación de la Propietaria del Producto, se sintió satisfecha de que el museo tuviese todas las obras y exposiciones que están en el museo real.
- La Propietaria del Producto estuvo de acuerdo con el nivel de dificultad de de la aplicación, pues no la consideró difícil de utilizar.

- Se acordó pequeños cambios en las luces para resaltar al modelo que en áreas estaba muy opaco.
- Se anunció la última iteración que integraría varios elementos multimedia al aplicativo ya desarrollado.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.4 ITERACIÓN III: COMPLEMENTOS MULTIMEDIA

3.4.1 Resumen

A fin de complementar el aplicativo ya desarrollado en la iteración anterior, se planteo el reto de expandir más la información contenida en el recorrido virtual. En esta iteración se agrega la base documental de las obras y el video del recorrido, además se genera un archivo .exe que no requiere instalación y la aplicación puede ser ejecutada desde el CD Multimedia sin necesidad de instalar la aplicación en el computador. Al final de la iteración el CD Multimedia fue entregado a la Propietaria del Producto para que lo pueda duplicar y distribuir como bien convenga, culminando así el desarrollo del aplicativo.

3.4.2 Reunión de Planificación del Sprint (Sprint Planning Meeting)

En esta reunión se seleccionaron los restantes que tenían que ver con los elementos multimedia. La Propietaria del Producto requería que se pueda acceder a la información de las pinturas sin ejecutar el recorrido, puesto que esto conlleva más tiempo. Para lo cual se plantea la integración de la base documental en una pantalla aparte de la aplicación, además de la integración del video de la iteración I y la compilación de todo el aplicativo en un CD Multimedia.

El esfuerzo estimado es menor que en las anteriores ocasiones, se define en 70 horas, dividida a lo largo de 14 días de trabajo.

Se estimó que se requeriría 10 horas para la creación de la base documental, 20 horas para el desarrollo de los estados de la aplicación, 30 horas para integrar la base documental en el aplicativo y 10 horas para compilar el CD Multimedia.

3.4.3 Pila de la Iteración (Sprint Backlog)

La pila de la iteración está compuesta de todos los requerimientos que estén relacionados con la integración de los elementos multimedia en la aplicación.

Tabla 3.15: Pila de la Iteración III.

Iteración III: Complementos Multimedia															
ÍTEM	Tarea	días de la iteración/esfuerzo restante													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
		70	64	55	48	44	43	39	29	24	18	14	10	4	0
10	Documentar la información de las obras y exposiciones.														
	Analizar la información obtenida en la Fundación Guayasamín.	2	0	0	0	0	0	0	0	0	0	0	0	0	
	Depurar la información de la base documental.	8	4	0	0	0	0	0	0	0	0	0	0	0	
11	Desarrollar los diferentes estados del aplicativo.														
	Definir los estados y opciones del aplicativo.	3	3	0	0	0	0	0	0	0	0	0	0	0	
	Generar el paso de estados.	8	8	6	0	0	0	0	0	0	0	0	0	0	
	Diseñar el entorno de cada estado.	7	7	7	6	2	1	0	0	0	0	0	0	0	
	Separar cada estado con sus respectivos métodos.	2	2	2	2	2	2	0	0	0	0	0	0	0	
12	Integrar la base documental en el aplicativo.														
	Cargar la información dentro de la aplicación.	4	4	4	4	4	4	4	0	0	0	0	0	0	
	Relacionar la información dentro del aplicativo.	8	8	8	8	8	8	7	2	0	0	0	0	0	
	Generar puntos de acceso a la información.	10	10	10	10	10	10	10	9	8	2	0	0	0	
	Generar la vista general de toda la base documental.	8	8	8	8	8	8	8	8	6	6	4	0	0	
13	Compilar toda la información en un medio óptico.														
	Generar el CD Multimedia.	10	10	10	10	10	10	10	10	10	10	10	10	4	0

En la siguiente sección se explica brevemente las actividades realizadas dentro de cada una de las tareas de la tabla 3.15.

3.4.4 Tareas

3.4.4.1 ÍTEM 10: Documentar la información de las obras y exposiciones.

- Analizar la información obtenida en la Fundación Guayasamín.
- Se realizó un análisis previo de toda la información proporcionada por la Fundación Guayasamín para definir lo necesario.
- Depurar la información de la base documental.
- Se resume en varios casos la información para que no sea tan extensa al momento de presentarla al usuario. El texto final que fue cargado en la base documental del aplicativo.

3.4.4.2 ÍTEM 11: Desarrollar los diferentes estados del aplicativo.

- Definir los estados y opciones del aplicativo.
- Para poder mostrar varias pantallas diferentes fue necesario desarrollar los estados de la aplicación y las opciones que tendrían cada uno de estos.
- Generar el paso de estados.
- Se desarrollaron los métodos requeridos para el paso de un estado a otro, como cuando se va del menú principal a la pantalla de los créditos.
- Diseñar el entorno de cada estado.
- Fue necesario diseñar una interfaz de usuario común pero con ciertos cambios para cada estado. La interacción de los estados se detalla en la clase "MenuInteraction".
- Separar cada estado con sus respectivos métodos.

- Para una mejor lectura del código fuente se separó los estados y sus métodos correspondientes. Los cambios se reflejan en el código fuente de la clase Game1.

3.4.4.3 ÍTEM 12: Integrar la base documental en el aplicativo.

- Cargar la información dentro de la aplicación.
- Es necesario el uso de varios arreglos de variables para poder indexar la información de la base documental y así poder acceder más fácilmente a ellos desde los diferentes métodos del programa.
- Relacionar la información dentro del aplicativo.
- Se utiliza variables comunes para los datos de tal modo que cuando se requiera acceder a ellos no se requiera la creación de instancias adicionales.
- Generar puntos de acceso a la información.
- Se desarrollan los métodos necesarios para que el usuario al dar clic con el mouse pueda visualizar la información de la obra seleccionada.
- Generar la vista general de toda la base documental.
- Se crea un estado para visualizar la base documental sin necesidad de entrar al recorrido virtual (Figura 3.28). Para poder acceder a la base documental se crea la clase “MuseumData”.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

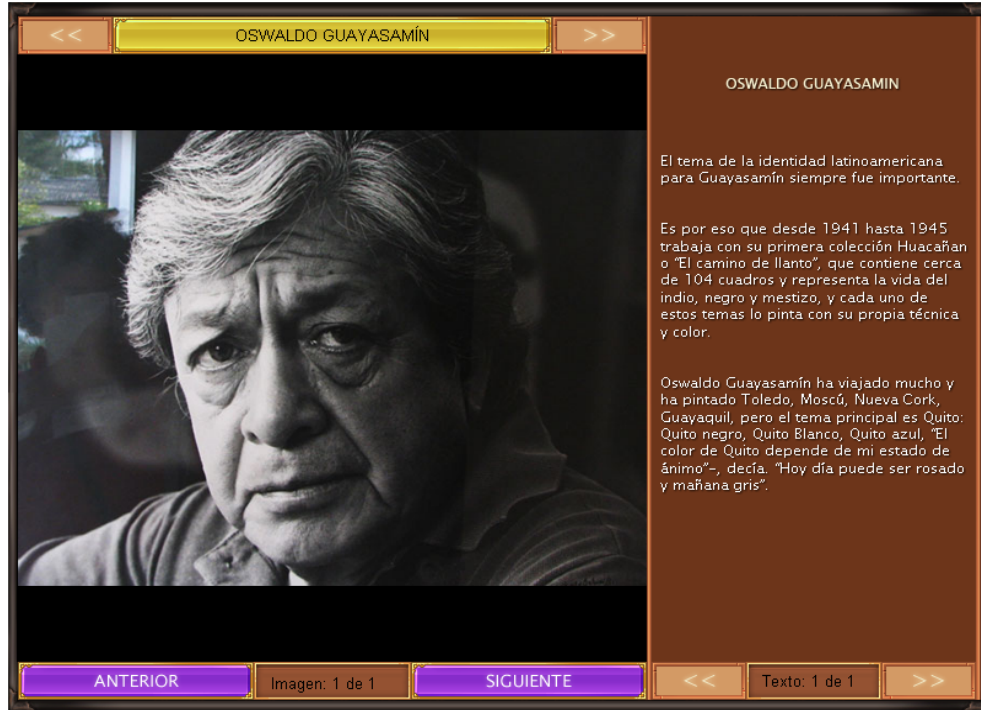


Figura 3.28: Captura de pantalla de la base documental.

3.4.4.4 ÍTEM 13: Compilar toda la información en un medio óptico.

- *Generar el CD Multimedia.*

Se compila la aplicación y los archivos extraídos y ordenados para que se pueda ejecutar la aplicación directamente desde el CD sin instalación.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

3.4.5 Documentos Generados

Al final de esta iteración se obtiene la versión final del Manual de Usuario (Anexo D), además de las versiones finales de cada una de las clases que se emplean en el aplicativo.

3.4.6 Gráfica de Trabajo Pendiente

En la figura 3.29 se puede observar el esfuerzo restante desde el inicio de la iteración en el día cero, hasta el final de la misma.

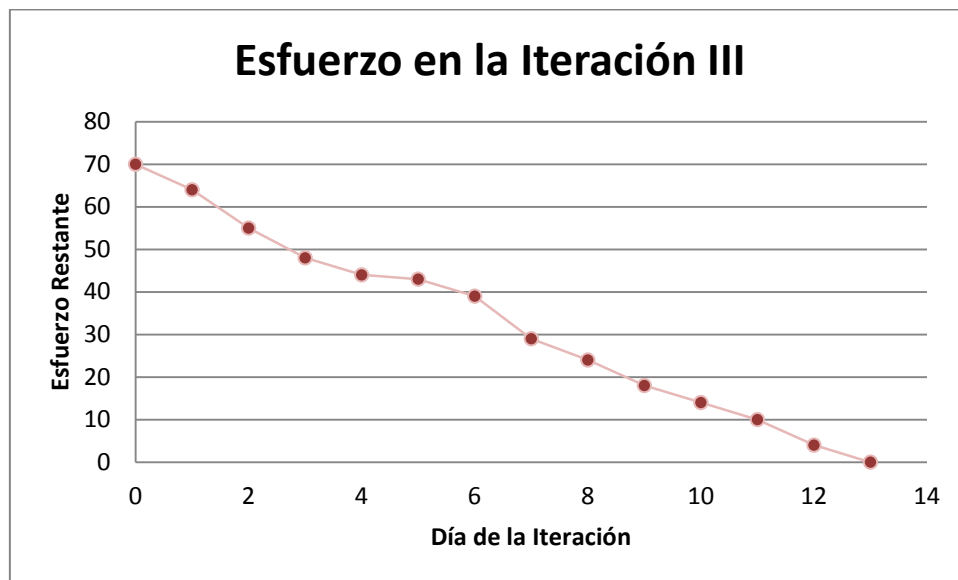


Figura 3.29: Gráfico de Trabajo Pendiente - Iteración III.

El gráfico muestra un leve estancamiento alrededor del sexto día, debido a que se presentaron dificultades al momento de integrar la base documental al aplicativo.

3.4.7 Reunión de Revisión de la Iteración (Sprint Review Meeting)

La reunión final marca la culminación del proyecto, para la entrega formalmente del producto completo, donde se determina lo siguiente:

- El aplicativo cumple con las expectativas de la Propietaria del Producto.
- Se procedió a evaluar la aplicación y no se encontró inconveniente al usarla.
- Se propuso el desarrollo de otras áreas de la fundación, proyectos que podrían ser desarrollados en otras tesis de grado.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

La metodología Scrum fue la indicada para el desarrollo del aplicativo, debido a que en cada iteración se pudo obtener un entregable funcional y verificable por el personal de la Fundación Guayasamín.

La herramienta 3ds Max 2010 permitió el correcto modelado del museo en un entorno 3D; las múltiples funciones que posee dieron la facilidad de obtener un modelo muy apegado a la realidad. La herramienta permite una rápida modificación del modelo en 3D para realizar cambios y adaptaciones, además de ser una herramienta muy versátil que permite modelar cualquier tipo de objeto geométrico.

Mediante el uso del Framework XNA el recorrido del museo obtuvo toda la interactividad y funcionalidad requerida para una correcta visualización por parte de los usuarios finales. XNA fue crucial para un correcto desarrollo del recorrido, permitiendo crear un aplicativo modificable de acuerdo a las necesidades presentes de movimiento, rapidez y accesibilidad.

El modelo virtual del museo de la Capilla del Hombre fue una gran oportunidad para realizar un producto capaz de mostrar, dentro del ámbito nacional e internacional, las obras del maestro Oswaldo Guayasamín. Las herramientas utilizadas dan una gran flexibilidad para adaptar el recorrido a los requisitos necesarios y fueron de vital importancia para que el producto final sea de fácil manejo y utilizable por cualquier persona.

En conjunto, todas las consideraciones tomadas para el desarrollo del Museo Virtual en 3D permitieron una acertada y oportuna entrega del producto final a la Fundación Guayasamín con todos los requisitos por ellos determinados.

ESPACIO DEJADO EN BLANCO INTENCIONALMENTE

4.2 RECOMENDACIONES

El museo virtual de La Capilla del Hombre puede ser complementado con la virtualización de las demás edificaciones administradas por la Fundación Guayasamín tales como la Biblioteca de Arte, la Casa Taller y el Museo de Arte Precolombino, Colonial y Contemporáneo. Por lo cual se recomienda a futuro una propuesta similar a la expresada en este proyecto.

Para una mayor difusión del modelo virtual generado, se recomienda a futuro una propuesta que enfoque el desarrollo a la Web, de tal modo que cualquier persona con una conexión de internet pueda acceder al recorrido virtual.

El modelo virtual se podría mejorar a futuro si se desarrolla un aplicativo para modificar de manera fácil y rápida la aplicación principal, lo que requerirá una persona encargada de la modificación y administración de cambios que podría tener más adelante el museo de la Capilla del Hombre.

Además se recomienda la creación de un recorrido virtual con gráficos más reales, para que sea una fiel representación del museo. No se llegó a un nivel alto de realismo en el recorrido debido a que se lo desarrolló para ser ejecutado en máquinas de bajos recursos.

En cuanto a la metodología, si se realiza un trabajo similar al expuesto en este documento, se recomienda el uso de estándares para la realización de pruebas de caja negra y caja blanca, en especial si se desea desarrollar un recorrido virtual en la web, ya que existiría un mayor número de ambientes en los cuales podría ejecutarse la aplicación y eso requeriría un mayor control del código desarrollado.

BIBLIOGRAFÍA

Albaladejo, X. (13 de Enero de 2009). *Proyectos Agiles: SCRUM*. Recuperado el 23 de Abril de 2011, de <http://www.proyectosagiles.org/>

Booch, G. (2006). *El lenguaje unificado de Modelado*. Addison wesley.

Conte, H. (2010). *SlideShare*. Recuperado el 24 de Marzo de 2011, de Proyecciones y Perspectivas: <http://www.slideshare.net/misio82/>

Foley, J. (1995). *Computer Graphics: Principles and Practice*. United States: Addison-Wesley.

Gregory, P. (2011). *CISA Certified Information Systems Auditor All-in-One Exam Guide*. Sacramento: McGraw-Hill.

Impagliazzo, J. (5 de Diciembre de 2004). Recuperado el 21 de Marzo de 2011, de Learning Computing History: http://www.comphist.org/computing_history/

Liu, J. (26 de Febrero de 2003). *The Newspaper of the University of Waterloo Engineering Society*. Recuperado el 14 de Abril de 2011, de http://iwarrior.uwaterloo.ca/props/?module=displaystory&story_id=1051

Münch, J. (2010). *New Modeling Concepts for Today's Software Processes*. Paderborn: Proceedings.

Pressman, R. S. (2006). *Ingeniería del software: un enfoque Práctico*. Madrid: McGraw-Hill.

Sevo, D. (2010). *History of computer graphics*. Recuperado el 14 de Marzo de 2011, de <http://www.danielsevo.com/hocg/>

BIOGRAFÍA

- **Datos Personales**

Nombre Completo: Luis Miguel Cóndor Guachamín

Fecha de Nacimiento: 8 de Agosto del 1986

Nacionalidad: Ecuatoriana

Estado Civil: Soltero

Dirección Domiciliaria: Avenida el Inca E11-23 y de las Toronjas

Teléfono Convencional: (02) 2417 076

Teléfono Celular: 08 6967680

Correo Electrónico: luis.condor.g@gmail.com

- **Formación Académica**

Primaria: Escuela Sangay

Secundaria: Colegio Blas Pascal - Físico Matemático

Superior: Escuela Politécnica del Ejercito

- **Idiomas**

Inglés: ESPE Instituto de Idiomas – Suficiencia en el idioma inglés.