



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Máster

DESARROLLO HERRAMIENTA DE DIFUSIÓN DE PATRIMONIO

Alumno/a: Gómez López, Alejandro

Tutor/a: Prof. D. Juan Roberto Jiménez Pérez
Dpto.: Informática

Julio, 2018

Contenido

Índice de Ilustraciones.....	6
Índice de Tablas	8
Resumen	9
Abstract.....	10
Confidencialidad	11
Agradecimientos	12
1. Introducción.....	14
1.1. La estructura del documento.....	14
1.2. La Digitalización 3D.....	15
1.3. Marco de Desarrollo	17
1.3.1. La empresa.....	17
1.3.2. El equipo.....	18
1.3.3. Tecnologías 3D para la Conservación y Difusión del Patrimonio	19
1.3.3.1. ¿En qué consiste?	19
1.3.3.2. ¿Por qué realizarla?.....	19
1.3.3.3. ¿Qué ventajas ofrece?.....	19
1.4. Motivación	22
UNIDAD I: LA DIGITALIZACIÓN 3D.....	24
2. Técnicas para escaneado y digitalización 3D en la actualidad	25
2.1. Digitalización por contacto	26
2.2. Digitalización sin contacto	27
2.2.1. Sistemas pasivos.....	27
2.2.1.1. Estereoscópicos.....	27
2.2.1.2. Silueta	27
2.2.2. Sistemas activos.....	28
2.2.2.1. Tiempo de vuelo.....	28
2.2.2.2. Triangulación.....	28
2.2.2.3. Luz estructurada	30
3. Obtención de modelos mediante escaneado y digitalización 3D.....	32
3.1. Creación del modelo en alta resolución	32
3.2. Creación del modelo con textura en alta definición	34
3.2.1. Fotogrametría: El método de reconstrucción	34
3.2.2. Equipo para fotogrametría.....	36
3.3. Modelado: “La Inmaculada Napolitana”	37
3.3.1. Modelos Obtenidos.....	42
UNIDAD II: EL PROCESO DE DESARROLLO.....	48
4. Análisis de requisitos	49
4.1. Alcance del proyecto	49
4.2. Principales requisitos del proyecto.....	50
4.3. Estructura de Desglose de Requisitos	51
4.4. Ciclo de Vida de la Gestión del Proyecto	52
4.4.1. ¿Por qué Scrum?.....	52
4.4.2. Adaptando Scrum.....	52
5. Elección de Tecnologías y Herramientas.....	54
5.1. Motores de Videojuego	54

5.1.1.	Unreal Engine	55
5.1.2.	Unity 3D	56
5.1.3.	CryEngine	58
5.2.	Motores de Renderizado	59
5.2.1.	Ogre 3D	59
5.3.	Comparativa	60
6.	Planificación	63
6.1.	Estructura de Desglose del Trabajo	63
6.2.	Estimación de la Duración	67
6.3.	Estimación de Recursos	67
6.3.1.	Personas.....	67
6.3.2.	Recursos Técnicos	67
6.3.2.1.	Infraestructuras	68
6.3.2.2.	Software	68
6.4.	Estimación de Costes.....	69
7.	Diseño Estructural.....	71
7.1.	Diagrama de Componentes del Sistema	71
7.1.1.	GUI.....	72
7.1.2.	GUI Manager	72
7.1.3.	Internalization Manager	72
7.1.4.	Mesh Serializer	72
7.1.5.	Zone Manager	72
7.1.6.	Control Manager	73
7.1.7.	Camera	73
7.1.8.	CrossPlatform Inputs	73
7.1.9.	Model Container	73
7.2.	Diagrama de Jerárquico de la Escena Principal	74
7.3.	Uso de patrones	74
8.	Ejecución	76
8.1.	Iteración 1.....	77
8.2.	Evaluación iteración 1	92
8.3.	Iteración 2.....	93
8.4.	Evaluación iteración 2	104
8.5.	Iteración 3.....	105
8.6.	Evaluación final	117
8.7.	Recursos usados durante el desarrollo.....	119
9.	Seguimiento y Control	121
9.1.	Comparativa entre Tiempo Estimado y Real por Iteraciones	121
9.2.	Comparativa entre Tiempo Estimado y Real por Tareas	123
9.3.	Coste y Fecha de Finalización	125
9.4.	Horas invertidas por tipo de Tarea	126
10.	Finalización del Proyecto	128
10.1.	Informe para el Equipo de Desarrollo.....	128
UNIDAD III: RESULTADOS Y CONCLUSIONES		131
11.	Exposiciones	132
11.1.	La obra invitada: Inmaculada Napolitana	132
11.2.	Catedral de Jaén	135
12.	Futuras Mejoras.....	140
13.	Conclusiones.....	142
UNIDAD IV: ANEXOS.....		144
14.	Anexo I: Uso de la Herramienta	145
15.	Anexo II: Manual de creación de proyectos.....	154

16. Referencias 162

Índice de Ilustraciones

<i>Ilustración 1 CMM Gama ALTERA de Nikon Metrology</i>	26
<i>Ilustración 2 Escáner de contacto Renishaw DS10 de alta precisión para uso en salud dental</i>	27
<i>Ilustración 3 Principio del método Tiempo de Vuelo</i>	28
<i>Ilustración 4 Principio del método Triangulación</i>	29
<i>Ilustración 5 Principio del método Luz Estructurada</i>	30
<i>Ilustración 6 Especificaciones técnicas del catálogo de escáneres InnovaSCAN</i>	32
<i>Ilustración 7 Software InnovaSCAN para el escáner de luz estructurada</i>	33
<i>Ilustración 8 Modelado de un busto usando fotogrametría con Agisoft PhotoScan, Nick Lievendag</i>	35
<i>Ilustración 9 Cámara Sony DSLR-A-380 usada para fotogrametría</i>	36
<i>Ilustración 10 Inmaculada Napolitana después del proceso de restauración 1</i>	37
<i>Ilustración 11 Inmaculada Napolitana después del proceso de restauración 2</i>	38
<i>Ilustración 12 Proceso Escáner de luz estructurada 1</i>	39
<i>Ilustración 13 Proceso Escáner de luz estructurada 2</i>	39
<i>Ilustración 14 Proceso Escáner de luz estructurada 3</i>	40
<i>Ilustración 15 Proceso Fotogrametría 1</i>	40
<i>Ilustración 16 Proceso Fotogrametría 2</i>	41
<i>Ilustración 17 Proceso Fotogrametría 3</i>	41
<i>Ilustración 18 Modelo a máxima resolución</i>	42
<i>Ilustración 19 Modelo en media resolución</i>	43
<i>Ilustración 20 Realidad frente a Modelo 1</i>	44
<i>Ilustración 21 Realidad frente a Modelo 2</i>	44
<i>Ilustración 22 Realidad frente a Modelo 3</i>	45
<i>Ilustración 23 Realidad frente a Modelo 4</i>	45
<i>Ilustración 24 Realidad frente a Modelo 5</i>	45
<i>Ilustración 25 Realidad frente a Modelo 6</i>	46
<i>Ilustración 26 Realidad frente a Modelo 7</i>	46
<i>Ilustración 27 Diagrama RBS del proyecto</i>	51
<i>Ilustración 28 Diagrama de Componentes del sistema</i>	71
<i>Ilustración 29 Árbol jerárquico de la escena 3D principal</i>	74
<i>Ilustración 30 Nuevo proyecto en Unity 3D</i>	78
<i>Ilustración 31 Escena principal construida</i>	79
<i>Ilustración 32 Material por defecto creado para los modelos</i>	81
<i>Ilustración 33 Cargado Rakan sin textura</i>	82
<i>Ilustración 34 Cargado Estatua Rakan con textura</i>	84
<i>Ilustración 35 Panel de botones de navegación de la GUI</i>	86
<i>Ilustración 36 Menú proyectos</i>	91
<i>Ilustración 37 Elemento punto y cylinder de los puntos de interés</i>	94
<i>Ilustración 38 Materiales de la geometría de los puntos de interés</i>	95
<i>Ilustración 39 Elemento cylinder y sphere del marcador de puntos de interés</i>	96
<i>Ilustración 40 Materiales del marcador de puntos de interés</i>	96
<i>Ilustración 41 Creando la geometría de un punto de interés</i>	97
<i>Ilustración 42 Punto de interés creado</i>	98
<i>Ilustración 43 Punto de interés seleccionado</i>	98
<i>Ilustración 44 Panel de Puntos de Interés básico</i>	99
<i>Ilustración 45 Nuevo panel de visualización de Punto de Interés</i>	101
<i>Ilustración 46 Creando Puntos de Interés 1</i>	102
<i>Ilustración 47 Creando Puntos de Interés 2</i>	103
<i>Ilustración 48 Añadiendo una imagen a un Punto de Interés</i>	103
<i>Ilustración 49 Mostrando la imagen anteriormente añadida</i>	104
<i>Ilustración 50 Unity Preferences</i>	106
<i>Ilustración 51 Cambio de Plataforma Objetivo</i>	107
<i>Ilustración 52 Configuración del proyecto Android</i>	108
<i>Ilustración 53 Fichero JSON "en_lang.json"</i>	111
<i>Ilustración 54 De izquierda a derecha, Panel de Opciones y Panel de Selección de Idiomas</i>	113
<i>Ilustración 55 Diferencia de intensidades sobre el modelo "Rakan"</i>	114

Ilustración 56 Panel de Selección de color, con diferentes variantes de color	116
Ilustración 57 Gráfico Porcentajes de satisfacción	119
Ilustración 58 Comparativa Tiempos Iteraciones	121
Ilustración 59 Diferencia Tiempos Iteraciones	122
Ilustración 60 Comparativa Tiempos Grupos de Tareas.....	123
Ilustración 61 Diferencia Tiempos Grupos de Tareas.....	124
Ilustración 62 Horas invertidas por tipo de Tarea	126
Ilustración 63 Cartel exposición "La obra invitada: Inmaculada"	132
Ilustración 64 Edificio Antiguo Magisterio de la Universidad de Jaén, lugar de la exposición .	133
Ilustración 65 Panfletos informativos del evento.....	134
Ilustración 66 Visitante manipulando el Modelo 3D	134
Ilustración 67 Día de la Inauguración.....	135
Ilustración 68 La Herramienta en la Catedral de Jaén 1.....	135
Ilustración 69 La Herramienta en la Catedral de Jaén 2.....	136
Ilustración 70 La Herramienta en la Catedral de Jaén 3.....	136
Ilustración 71 La Herramienta en la Catedral de Jaén 4.....	137
Ilustración 72 La Inmaculada Napolitana restaurada en la Catedral de Jaén	138
Ilustración 73 Interfaz de Usuario.....	145
Ilustración 74 Panel Selección de Proyectos	146
Ilustración 75 Panel de Opciones.....	146
Ilustración 76 Panel selección modelos secundarios.....	147
Ilustración 77 Creando un Punto de Interés.....	147
Ilustración 78 Panel para completar información de un punto de interés	148
Ilustración 79 Añadiendo Imágenes a un punto de interés	149
Ilustración 80 Punto de interés seleccionado, con un marcador	149
Ilustración 81 Información asociada a un punto de interés.....	150
Ilustración 82 Panel de Navegación.....	151
Ilustración 83 Carpeta "Import" para la importación del Modelo	154
Ilustración 84 Configuración de importación de Textura.....	155
Ilustración 85 Asociación de Textura al material.....	155
Ilustración 86 Modelo de Pruebas siendo posicionado en la escena	156
Ilustración 87 Cubo en el Origen que sirve como guía	156
Ilustración 88 Al pulsar "Play" comprobamos la correcta posición	157
Ilustración 89 Script Complex Mesh Serializer.....	158
Ilustración 90 Log del sistema al crear el proyecto	158
Ilustración 91 Lista de proyectos existentes en el sistema	159
Ilustración 92 Carpeta donde se encuentran los modelos secundarios de un proyecto	160
Ilustración 93 Modelo base y modelo secundario de ejemplo	160

Índice de Tablas

<i>Tabla 1 Requisitos principales extraídos de la descripción del sistema</i>	<i>50</i>
<i>Tabla 2 Comparativa Tecnologías (27/02/2017).....</i>	<i>60</i>
<i>Tabla 3 WBS del proyecto.....</i>	<i>65</i>
<i>Tabla 4 Estimación Temporal por grupos de tareas</i>	<i>66</i>
<i>Tabla 5 Iteraciones.....</i>	<i>67</i>
<i>Tabla 6 Infraestructuras requeridas para el proyecto</i>	<i>68</i>
<i>Tabla 7 Software requerido para el proyecto</i>	<i>68</i>
<i>Tabla 8 Resumen de Costes.....</i>	<i>69</i>
<i>Tabla 9 Cuestionario de Evaluación Final.....</i>	<i>118</i>
<i>Tabla 10 Resumen Porcentajes de satisfacción</i>	<i>118</i>
<i>Tabla 11 Resumen de Tiempos de las Iteraciones.....</i>	<i>121</i>
<i>Tabla 12 Resumen de Tiempos de los Grupos de Tareas</i>	<i>123</i>
<i>Tabla 13 Fechas Estimadas de las Iteraciones</i>	<i>125</i>
<i>Tabla 14 Fechas Reales de las Iteraciones</i>	<i>125</i>

Resumen

Este trabajo fin de máster trata sobre el desarrollo de una herramienta de difusión de patrimonio, capaz de enriquecer modelos 3D escaneados con información multimedia.

La idea nace de la necesidad de reforzar y completar servicios de digitalización y modelado 3D ofrecidos por una empresa. Este sistema permite que visitantes de cualquier edad manipulen libremente el modelo 3D de una pieza artística y vayan descubriendo interactivamente la información que ocultan sus detalles. Se exponen detalladamente los procesos reales de análisis, diseño, implementación y pruebas llevados a cabo durante la fase de desarrollo.

Como paso previo se realiza un repaso por las técnicas más actuales sobre modelado 3D y se muestra el flujo de trabajo seguido en la empresa para la obtención de modelos de alta resolución.

Finalmente se presentan unas exposiciones donde el público ha podido usar el prototipo y unas conclusiones propias del autor sobre el proyecto en general.

Abstract

This master's thesis is about the development of a cultural heritage dissemination tool, capable of enriching scanned 3D models with multimedia information.

The idea comes from the need to reinforce and complete 3D scanning and modeling services offered by a company. This system will allow visitors of any age to freely manipulate the 3D model of an artistic piece and interactively discover the information that conceals its details. The real processes of analysis, design, implementation and tests carried out during the development phase are detailed.

As a previous step, a review is made of the most current 3D modeling techniques and the workflow followed in the company to obtain high resolution models is shown.

Finally, some exhibitions are presented where the public has been able to use the prototype followed by the author's own conclusions about the project in general.

Confidencialidad

El proyecto en el que se basa este trabajo fin de máster es propiedad intelectual de 4D Geoservices S.L. La empresa no autoriza su difusión completa o parcial ni su publicación en abierto por ningún medio, ya sea físico o digital.

Agradecimientos

A 4D Geoservices, que me dio la oportunidad.

A mi tutor, que me guió.

A mi familia, que me aguantó.

A mi chica, que me apoyó.

1. Introducción

Los grandes avances realizados en digitalización e impresión 3D en los últimos años, junto con el abaratamiento de los dispositivos hardware necesarios para su aplicación han propiciado el uso de estas tecnologías a nivel empresarial. Cualquier organización, independientemente de su tamaño, tiene la posibilidad de mejorar sus servicios o crear nuevas líneas de trabajo gracias a una modesta inversión, obteniendo resultados de gran calidad. El carácter transversal de estas tecnologías y la constante aparición de nuevos campos de aplicación hacen pensar que sólo la imaginación marca el límite.

Éste es el caso concreto de la empresa, 4D Geoservices S. L. , con sede en Jaén capital. 4D Geoservices posee líneas de trabajo estrechamente relacionadas con el modelado y escaneado 3D. Con el fin de reforzar estos servicios y desmarcarse de sus competidores nace la idea de este proyecto.

Este trabajo fin de máster trata sobre el desarrollo de una herramienta de difusión de patrimonio, capaz de enriquecer los modelos 3D escaneados con información multimedia. Este sistema brindará a los usuarios una nueva forma de visita virtual, dejando que éstos manipulen libremente la representación virtual en 3D de una pieza artística y vayan descubriendo interactivamente la información que ocultan sus detalles. Como paso previo se realizará un repaso por las técnicas más actuales sobre modelado 3D y se mostrará el flujo de trabajo seguido en la empresa para la obtención de modelos de alta resolución. Se expondrán detalladamente los procesos reales de análisis, diseño, implementación y pruebas llevados a cabo durante la fase de desarrollo. Finalmente se presentarán los resultados obtenidos, las exposiciones donde se ha usado el prototipo y unas conclusiones propias del autor sobre el proyecto en general.

A continuación se muestra la forma en la que está organizado este documento, acompañada de una breve introducción a la digitalización 3D y el marco de desarrollo. Estos dos últimos apartados son de gran importancia, ya que familiarizarán al lector con la naturaleza del proyecto y las particulares condiciones que han impulsado su desarrollo desde la empresa.

1.1. La estructura del documento

Este documento está organizado en 4 grandes unidades. La primera de ellas (la digitalización 3D) trata sobre las actuales técnicas de modelado 3D y la aplicación de éstas a casos reales. A través de la fotogrametría y un escáner de luz estructurada la empresa genera modelos de alta resolución para diferentes fines. Se dará a conocer los detalles de este trabajo y las herramientas usadas.

La segunda unidad (el proceso de desarrollo) engloba la totalidad del proceso del desarrollo software, desde un temprano análisis de requisitos hasta

la generación y entrega de los informes y versiones finales del prototipo. Esta unidad representa secuencialmente el proceso real seguido por el equipo de la empresa a la hora de afrontar este proyecto.

La tercera unidad (resultados y conclusiones) presenta las exposiciones en las que ha estado presente el prototipo desde su desarrollo y unas conclusiones generales propias del autor sobre este trabajo a modo de cierre.

Finalmente, la cuarta unidad (Anexos) recoge los dos documentos generados durante el proceso de desarrollo: El manual de uso y el manual de creación de proyectos.

1.2. La Digitalización 3D

Digitalizar consiste en transformar una medición analógica de una magnitud en un valor numérico. Se define la digitalización 3D como el proceso por el cual se transforma una medición analógica de las dimensiones espaciales de un objeto en un conjunto de coordenadas numéricas (X,Y,Z). Las técnicas del escaneado y digitalización 3D capturan automáticamente una gran cantidad de puntos para definir con la más alta fidelidad posible la forma del objeto.

La digitalización 3D ha sufrido una serie de importantes avances tecnológicos. Cada vez es posible digitalizar objetos con formas más complejas y de diversos materiales con mayor precisión. La digitalización y la impresión 3D (también conocida como impresión aditiva) no sólo tienen un considerable peso en la industria de la producción, sino que también aportan grandes beneficios a otros campos como:

- Atención médica: La industria médica utiliza escáneres 3D para digitalizar el cuerpo humano completo, o sus partes, con el fin de llevar a cabo investigaciones. Las compañías también utilizan los datos digitalizados para desarrollar equipos que se adapten mejor al cuerpo humano.
- Educación e Investigación: Los profesores pueden utilizar los escáneres 3D y las impresoras 3D para instruir mejor a sus alumnos en diseño industrial, diseño mecánico, ingeniería y muchas otras áreas. Los investigadores pueden aprovechar la digitalización 3D y la impresión 3D para analizar objetos físicos.
- Multimedia: Los artistas que realizan trabajos 3D en computadoras utilizan la digitalización 3D para producir simulaciones

extremadamente realistas del mundo real, ya sea para videojuegos o con fines militares, científicos, educativos o de entrenamiento.

- Conservación y Difusión del patrimonio: Los arqueólogos y los investigadores pueden digitalizar objetos y artefactos para archivarlos e investigarlos. También pueden imprimir modelos 3D para exhibirlos en todo el mundo.

- Arte y Arquitectura: Los artistas, escultores y arquitectos pueden digitalizar objetos artísticos o elementos de un diseño arquitectónico. Pueden realizar cambios en el diseño de forma digital y, luego, imprimirlos nuevamente para ver los resultados.

Estos campos de aplicación son sólo ejemplos de una larga lista que, día a día va en aumento (Kurman & Lipson, 2014).

La digitalización y la impresión 3D crean un “puente” bidireccional entre el mundo real y el digital. Prácticamente cualquier objeto físico puede digitalizarse con eficiencia y crearse de forma digital obteniendo los niveles más altos de calidad y realismo. Estos datos resultantes pueden utilizarse para infinidad de objetivos tales como el desarrollo de nuevos productos, rápida creación de prototipos, ingeniería inversa, mejora de componentes existentes, investigación o salvaguarda de información importante y exclusiva (Jorquera Ortega, 2017).

La digitalización de un objeto es dependiente del modelo que se desee obtener. Se utilizarán unas técnicas y tecnologías según la naturaleza del objeto (materiales, zonas transparentes,...) y el resultado final esperado. Algunas las técnicas son muy rápidas pero su área de acción es limitada, haciendo inviable su uso (en tiempo o coste) para la digitalización de grandes objetos. Otras técnicas no toleran superficies transparentes o digitalizaciones en exteriores.

Estas metodologías se encuentran divididas en dos grandes campos, atendiendo al nivel invasivo que presenten para con el objeto a escanear. Existen métodos por contacto físico y otros en los que no es necesario.

Los métodos por contacto suelen estar compuestos por un sensor táctil movido por un brazo mecánico o un sistema de posicionamiento. Cada punto se determina manteniendo el contacto físico con el objeto real. Además, son sistemas de gran precisión, lentos en el proceso de digitalización y muy caros. El contacto físico descarta estos métodos para su uso en conservación del patrimonio artístico cultural.

1.3. Marco de Desarrollo

A principios de 2017 el alumno inició una relación laboral como ingeniero informático mediante un contrato en prácticas con una empresa llamada 4DGeoservices, alojada en el vivero del Edificio Antiguo Magisterio, en Jaén. Actualmente se continúa formando parte de la empresa.

Durante este tiempo se ha trabajado en proyectos referentes a diversos campos. Algunos de ellos relacionados con aerotriangulación, sistemas SIG, modelado 3D, sistemas de tracking por GPS y seguridad ciudadana.

Este trabajo fin de máster refleja un proyecto en el que el alumno ha estado activamente involucrado desde sus inicios, dando como resultado un producto real, diseñado, desarrollado y utilizado en un entorno empresarial real y expuesto directamente al público.

A continuación se presentará la empresa, las personas que la componen y las líneas de trabajo que están fuertemente relacionadas con el proyecto.

1.3.1. La empresa



4DGeoservices S.L. es una empresa con sede en Jaén que nace de la experiencia de profesionales relacionados con la captura, análisis y gestión de información geográfica. Su actividad se centra en proporcionar soluciones para la gestión inteligente del territorio, del patrimonio y de edificios mediante la integración de diferentes herramientas basadas en el uso de técnicas y tecnologías geomáticas (4D Geoservices, 2017).

4DGeoservices se apoya en técnicas de última generación para aportar respuestas a aquellas necesidades que requieran soluciones innovadoras de posicionamiento o digitalización, integrando tecnologías avanzadas y elaborando sus propias líneas de investigación y desarrollo.

Las principales líneas de trabajo que ofrece 4DGeoservices se estructuran básicamente en 7 grandes bloques:

- Posicionamiento en interiores (GeoIndoor)
- Modelado e impresión 3D
- Fotogrametría
- SIG y Servicios Web Geoespaciales
- Servicios Basados en la Localización (LBS)
- Cartografía y Topografía
- Realidad Aumentada

1.3.2. El equipo

El equipo se compone de los dos socios fundadores y el autor de éste trabajo. Los perfiles profesionales son:

- **JPC:** Ingeniero en Geodesia y Cartografía y Máster en Tecnologías Geoespaciales para la Gestión Inteligente del Territorio. Fundador de 4DGeoservices, experto en Geomática, especialmente en la captura, modelado y análisis de información 3D en proyectos de investigación. Integrando soluciones de posicionamiento con información contextual.
- **CC:** Ingeniero Técnico en Topografía, Graduado en Ingeniería Geomática y Topográfica y elaborando el TFM del Máster de Tecnologías Geoespaciales para la Gestión Inteligente del Territorio. Fundador de 4DGeoservices, empresa de soluciones geomáticas. Desarrollo de soluciones de posicionamiento en interiores para smartphones.
- **AG:** Graduado en Ingeniería Informática, especialización en Ingeniería del Software, y cursando el Máster en Ingeniería Informática (UJA). Autor de este trabajo fin de Máster. Responsable de diseño y desarrollo de aplicaciones empresariales y soluciones "Ad-hoc" en entornos distribuidos y web. Responsable de la gestión de servicios de tecnologías de la información.

1.3.3. Tecnologías 3D para la Conservación y Difusión del Patrimonio

1.3.3.1. ¿En qué consiste?

El digitalizado 3D es una técnica usada para la obtención de la geometría de objetos tridimensionales a partir de estudios fotográficos o mediante el escaneo 3D. En el ámbito de la ingeniería digital religiosa, el proceso de digitalización 3D permite la obtención de copias de seguridad. Éstas pueden equipararse a la realización de una réplica o molde virtual que captura por completo la superficie de la imagen. Su resolución es tal que cualquier detalle de la misma queda registrado para su futuro uso en múltiples aplicaciones de los modelos tridimensionales (4D Geoservices, 2017).

1.3.3.2. ¿Por qué realizarla?

Una de las principales preocupaciones que existe dentro del seno de las cofradías es la de conservar, en el mejor estado posible, su patrimonio artístico para que sea transmitido a las futuras generaciones.

Son numerosas las cofradías que han optado por la realización de copias de seguridad de sus titulares para protegerlos ante posibles accidentes y deterioros, siendo el digitalizado en 3D una tendencia mundial en la conservación del patrimonio histórico-cultural.

Los modelos en 3D generados son idóneos para la documentación patrimonial y constituyen una copia de seguridad de gran calidad métrica, con precisiones submilimétricas, que pueden ser usadas para la reproducción exacta de la imagen en caso necesario (4D Geoservices, 2017).

1.3.3.3. ¿Qué ventajas ofrece?



Totalmente inocuo para la talla

En el estudio realizado, en ningún momento se entra en contacto con la superficie policromada, por lo que se evitan riesgos de deterioro de la misma.



Difusión

Los modelos creados pueden ser incrustados en documentos PDF o visores Web para su divulgación al público amante de la Semana Santa y del arte en general.



Rapidez y comodidad

Los trabajos se realizan en el lugar donde se encuentre la imagen expuesta al público, reduciendo riesgos en traslados.



Documentación 3D

El patrimonio artístico de una cofradía queda documentado de manera digital.



Copias de Seguridad

A partir del Modelo 3D digital, y mediante impresoras 3D o maquinaria de tallado numérico, pueden realizarse reproducciones totalmente fieles al original, ya sea de la talla completa o de determinados elementos y a cualquier escala.



Fácil custodia

El modelo 3D digital conlleva una custodia tan sencilla y económica como el uso de un disco duro frente a las copias físicas realizadas tradicionalmente que requieren de un considerable espacio. Ésta tecnología ofrece una total confidencialidad en el caso que así se requiera.

1.4. Motivación

Como se ha podido comprobar en los puntos anteriores, una parte de las actividades de la empresa se centra en la generación de modelos 3D de alta resolución. Con el fin de brindar a los clientes servicios extra que diferencien a la empresa del resto de competidores, se investigaron nuevas necesidades que no estaban totalmente satisfechas en este nicho de mercado.

El proyecto surge de la doble necesidad de:

- Servir como herramienta de difusión de patrimonio, brindando a visitantes de exposiciones información de manera interactiva de una obra de arte. La idea consiste en tener la aplicación en un pantalla táctil en la que los usuarios puedan manipular un modelo 3D de la obra en alta resolución y consultar información multimedia (videos, audios, imágenes y textos) asociada a la obra en general o a partes concretas de ésta.

- Servir como herramienta de conservación de patrimonio, optimizando el trabajo de los restauradores. Actualmente el proceso de restauración de una obra genera una ingente cantidad de documentos y fotografías, haciendo difícil y tediosa la tarea de gestionar toda esta información. Esta versión de la aplicación permitiría gestionar todos los cambios geométricos y de apariencia (texturas) sufridos por la obra a lo largo del proceso de restauración con toda la información e informes asociados a los mismos en un único proyecto, siendo capaz de generar la documentación final requerida por los restauradores.

UNIDAD I: LA DIGITALIZACIÓN 3D

*“La digitalización es implacable, algunos lo entenderán
y el resto serán sustituidos biológicamente.”*

Genís Roca

2. Técnicas para escaneado y digitalización 3D en la actualidad

La captación en 3D de una escena ha sido uno de los grandes objetivos de la visión artificial. Tratando de conseguir este objetivo de la forma más eficiente posible surgieron los métodos sin contacto, activos y pasivos. Los métodos pasivos se basan en puntos característicos presentes en 2 o más imágenes. En contraposición, los métodos activos, como la luz estructurada, parten de la generación de características sobre el objeto a través de un emisor de luz.

Una de las primeras técnicas (hace más de cuarenta años) nació de la necesidad de captar escenas por los mecanismos de visión artificial. Esta técnica consistía en la reconstrucción tridimensional de escenas gracias a un par estereoscópico de cámaras, intentando imitar la visión humana.

Este método intentaba simular el efecto de la estereoscopia, descubierto por Sir Charles Wheatstone en 1840. Resultó ser una gran aproximación al problema pero no una solución definitiva. La visión humana usa otros factores, como la propia experiencia por ejemplo para captar el entorno. Cabe mencionar que este método sigue siendo el más usado en visión artificial ya que su carga de procesamiento es la más baja con respecto a otros dispositivos de captación 3D (Gagalowicz & Philips, 2011).

A grandes rasgos, el proceso de captación de la escena tridimensional mediante esta técnica es el siguiente:

- Calibración: Obtención de los parámetros intrínsecos y de distorsión de cada cámara individualmente.
- Correspondencia: Identificación de la proyección correspondiente en la imagen contraria.
- Reconstrucción: Cálculo de la coordenada espacial a partir de la disparidad en las proyecciones.

Los escáneres estereoscópicos usan esta técnica y son sistemas de captación pasivos. Este tipo de técnica ha dado pie a la aparición de otras basadas en ella, como la usada por los escáneres de silueta.

Si bien el descubrimiento de la estereoscopia se remonta a 1840, su uso en computación es mucho más reciente. El primer sistema de captación 3D que realmente generaba un producto de calidad (evidentemente muy inferior a la calidad de los sistemas actuales) fue el método de captura por contacto (Lahoz, Calderón, & Aguilera, 2010).

En sus inicios fué el sistema que dió los mejores valores de precisión. Una punta de acero se desplazaba por la superficie del objeto. Este proceso fue mejorando el tiempo requerido, obteniendo muy buenos resultados. Su precisión es muy alta pero sigue considerándose un proceso muy lento comparado con otras técnicas como la tecnología laser o la luz estructurada.

A día de hoy existen dos tipos de escáneres 3D atendiendo al hecho de que sea necesario el contacto físico con el objeto o escena en cuestión. Además, los escáneres sin contacto pueden subdividirse en dos categorías principales, escáneres activos y pasivos.

2.1. Digitalización por contacto

Estos sistemas examinan el objeto apoyando sobre la superficie de éste el elemento de medida (habitualmente llamado palpador). Este elemento suele ser una punta de acero, rubí, óxido de circonio o zafiro, entre otros. Unos sensores internos permiten determinar la posición del palpador en el espacio. Son ampliamente usados en procesos de fabricación y pueden alcanzar precisiones de hasta 0,01 mm. Las mayores desventajas son la necesidad del contacto físico con el objeto (sobre todo si éste es frágil o valioso) y la lentitud del proceso de escaneado en comparación con otras técnicas que serán vistas a continuación.

Ejemplos de estos sistemas son las CMM (Máquina de medición de coordenadas) o escáneres de contacto con brazos robóticos.



Ilustración 1 CMM Gama ALTERA de Nikon Metrology



Ilustración 2 Escáner de contacto Renishaw DS10 de alta precisión para uso en salud dental

2.2. Digitalización sin contacto

2.2.1. Sistemas pasivos

Los escáneres pasivos no emiten ninguna clase de radiación por sí mismos, detectan una radiación ya existente. La mayoría de los escáneres de este tipo detectan la luz visible puesto que es una radiación disponible en el ambiente. Los métodos pasivos suelen ser muy baratos, porque en la mayoría de los casos no necesitan un hardware específico.

2.2.1.1. Estereoscópicos

Los sistemas estereoscópicos utilizan el mismo principio de la fotogrametría, utilizando la medida de la paralaje entre dos imágenes para determinar la distancia de cada pixel de la imagen. Emplean generalmente dos cámaras de video, levemente separadas, mirando a la misma escena. Analizando las diferencias leves entre las imágenes vistas por cada cámara, es posible determinar la distancia en cada punto en las imágenes. Este método se basa en la visión estereoscópica humana (Gagalowicz & Philips, 2011).

2.2.1.2. Silueta

Estos tipos de escáneres 3D usan bosquejos creados de una sucesión de fotografías alrededor de un objeto tridimensional contra un fondo muy bien contrastado. Estas siluetas se estiran y son cruzadas para formar la aproximación visual de casco del objeto. Con esta clase de técnicas algunas concavidades de un objeto (como el interior de un tazón) no son detectadas.

2.2.2. Sistemas activos

Los escáneres activos emiten alguna clase de señal y analizan su retorno para capturar la geometría de un objeto o una escena. Se utilizan radiaciones electromagnéticas (desde ondas de radio hasta rayos X) o ultrasonidos.

2.2.2.1. Tiempo de vuelo

Un escáner 3D de tiempo de vuelo determina la distancia a la escena cronometrando el tiempo del viaje de ida y vuelta de un pulso de luz. Un diodo láser emite un pulso de luz y se cronometra el tiempo que pasa hasta que la luz reflejada es vista por un detector. Como la velocidad de la luz C es conocida, el tiempo del viaje de ida y vuelta determina la distancia del viaje de la luz, que es dos veces la distancia entre el escáner y la superficie. Si T es el tiempo del viaje completo, entonces la distancia es igual a $(C * T)/2$. Claramente la certeza de un escáner láser de tiempo de vuelo 3D depende de la precisión con la que se puede medir el tiempo T : 3,3 picosegundos (aproximadamente) es el tiempo requerido para que la luz viaje 1 milímetro. Se utilizan láseres visibles (verdes) o invisibles (infrarrojo cercano).

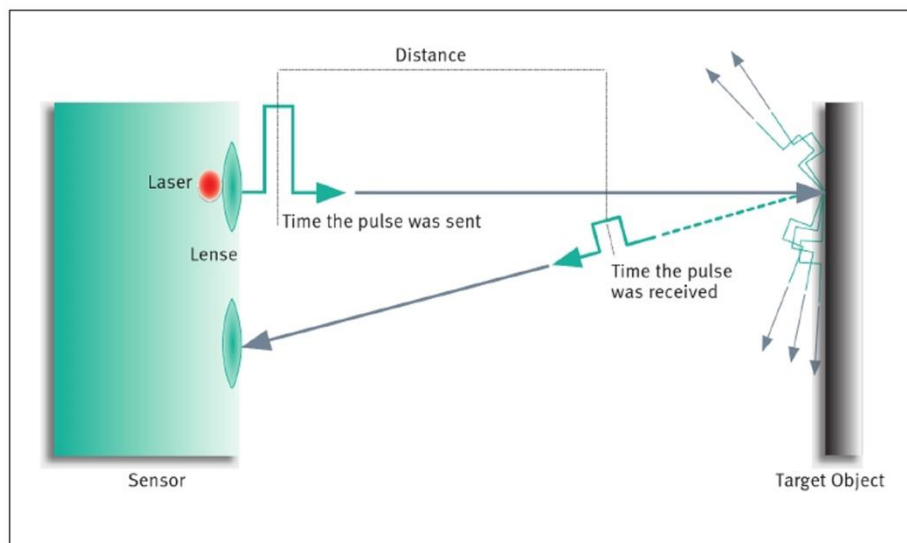


Ilustración 3 Principio del método Tiempo de Vuelo

2.2.2.2. Triangulación

El escáner láser de triangulación 3D es también un escáner activo que usa la luz del láser para examinar el entorno. El haz de luz láser incide en el objeto y se usa una cámara para buscar la ubicación del punto del láser. Dependiendo de la distancia a la que el láser golpee una superficie, el punto del láser aparece en lugares diferentes en el sensor de la cámara.

Esta técnica se llama triangulación porque el punto de láser, la cámara y el emisor del láser forman un triángulo. La longitud de un lado del triángulo definido por la cámara y el emisor del láser es conocida. El ángulo del vértice del emisor de láser se sabe también. El ángulo del vértice de la cámara (paralaje) puede ser determinado mirando la ubicación del punto del láser en la cámara. Estos tres valores permiten determinar el resto de las dimensiones del triángulo, y por tanto, la posición de cada punto en el espacio.

La precisión de este sistema de medida puede ser muy elevada (milésimas de milímetro), pero depende del ángulo del vértice opuesto al escáner (cuanto más se aparte de 90° más baja es la precisión), lo que limita el tamaño de la escena a analizar. Dado que ese ángulo depende fuertemente de la distancia entre el emisor láser y la cámara, el aumentar el alcance supone incrementar mucho el tamaño del equipo de medida. En la práctica, el alcance máximo de estos escáneres se limita a 20-30 cm.

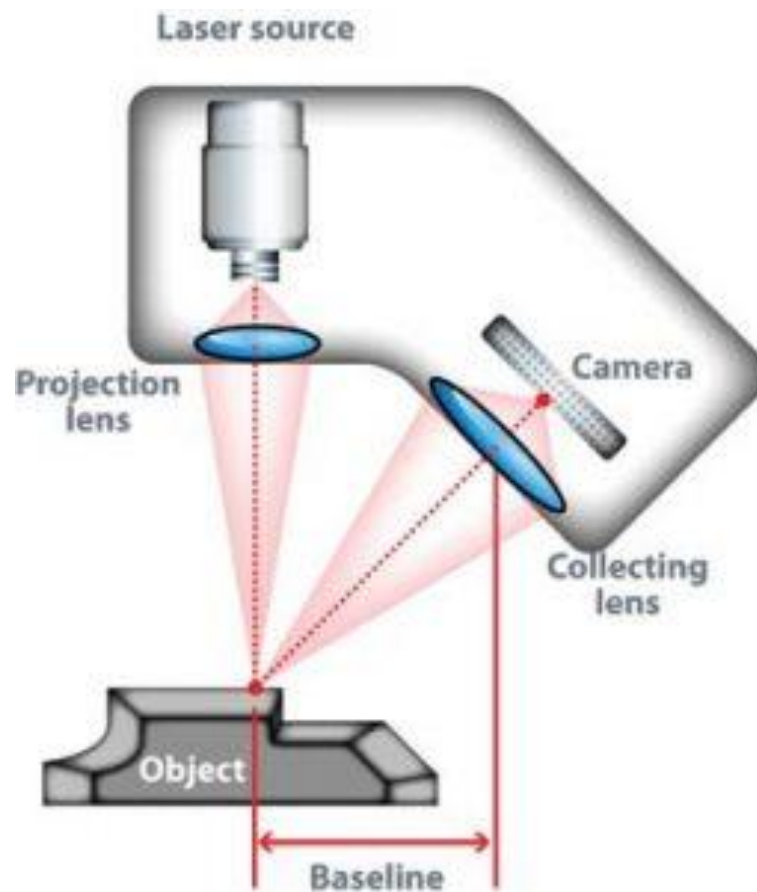


Ilustración 4 Principio del método Triangulación

2.2.2.3. Luz estructurada

Los escáneres 3D de luz estructurada proyectan un patrón de luz generado en el objeto y analizan la deformación del patrón producida por la geometría de la escena. El patrón de luz suele consistir en un conjunto de líneas paralelas generadas bien por interferencia laser o por proyección. El reflejo se captura con una cámara fotográfica y posteriormente mediante unos algoritmos se determina la posición de cada punto en el espacio. En algunos casos, dos cámaras fotográficas a los lados del emisor de luz proporcionan mejores resultados.

Como con el resto de tecnologías ópticas, este tipo de escáneres tienen problemas con las superficies transparentes y reflexivas puesto que la luz no interfiere en ellas el mismo reflejo que en las opacas. Una manera de solucionar este problema es aplicando una capa fina de laca opaca a las superficies problemáticas.

La ventaja de los escáneres 3D de luz estructurada es la velocidad. En vez de escanear un punto a la vez, escanean múltiples puntos o el campo entero del panorama inmediatamente. Esto reduce o elimina el problema de la deformación del movimiento. Algunos sistemas existentes son capaces de escanear objetos en movimiento en tiempo real.

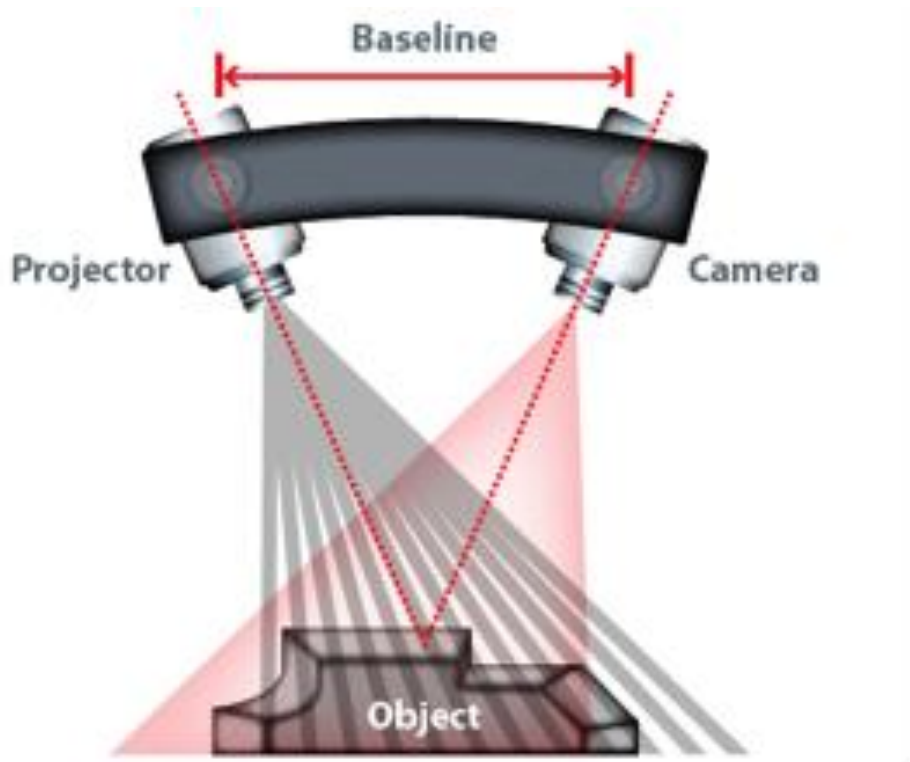


Ilustración 5 Principio del método Luz Estructurada

3. Obtención de modelos mediante escaneado y digitalización 3D

Según el flujo de trabajo usado en la empresa, debemos distinguir dos tareas. La creación de un modelo (sin textura) de alta resolución usando un escáner de luz estructurada y la creación de un modelo con menor resolución pero con una textura en alta definición, usando técnicas de fotogrametría con una cámara réflex.

3.1. Creación del modelo en alta resolución

La empresa cuenta con un escáner de luz estructurada marca [InnovaSCAN](#), modelo "Light", consistente en un par de cámaras fotogramétricas y un proyector. InnovaSCAN tiene otros modelos (tal y como se aprecia en la [Ilustración 6](#)) con mayores prestaciones y un precio superior.




specification	model	HEAVY DUTY QUADRO	HEAVY DUTY	PRO+	LOUPE+	LIGHT
Light-source type		R/G/B	R/G/B	White	White	White
Number and type of cameras		4 x 5 Mpx	2 x 5 Mpx	2 x 5 Mpx	2 x 5 Mpx	2 x 1,3 Mpx
Scanning accuracy according to DE VDI/VDE2634 Part 2, 4.1 Ps		from 0.01 mm		from 0.025 mm	from 0.01 mm	from 0.04 mm
Scanning time		5 sek.		4 sek.		
Measuring ranges		230x190x120 mm and 400x330x160 mm (1000 x 800 x 250 mm optional)	230x190x120 mm and 400x330x160 mm (1000 x 800 x 250 mm optional)	max. (430 x 330 mm), min. (300 x 230 mm), 7 ranges	max. 460 x 350 mm min. 100 x 80 mm 15 ranges	max. 520 x 420 mm min. 380 x 300 mm 7 ranges
Export formats		stl, ply, obj, asc, bin				
Points density		115 pkt/mm ² [narrow range] i 43 pkt/mm ² [wide range]	115 pkt/mm ² [narrow range] or 43 pkt/mm ² [wide range]	max. 73 / min. 36	max. 625 / min. 31	max. 11/ min. 6
Computer connection		1x USB 3.0 or 3x Gigabit Ethernet	1x USB 3.0 or 3x Gigabit Ethernet	1 x USB 3.0 or 2 x Gigabit Ethernet, HDMI/VGA/DVI		USB HDMI/VGA/DVI
Hardware requirements		Windows 7 (64-bit), 16 GB RAM, CPU i5				Windows 7 (64-bit), 4 GB RAM, CPU i5
Software		eviXscan				
Dimensions [mm]						
Scanner		500x245x95		740x400x200	470x470x200 (without wings) 870x470x200 (with wings)	740x400x200
Scanner on tripod		1000x1000x1000				
Complete set		ca. 600x300x280		820x440x240	920x550x260	820x440x240
Weight [kg]						
Scanner		7		13	14 (without wings) 16 (with wings)	13
Set		10 (case excluded)		22	30	22
Temperature						
Ambient operation temperature		from +10°C to +30°C				
Storage temperature		from -20°C to +40°C				
Electrical						
AC input		110/230 V, 2,5 A, 50/60 Hz				
International Protection Rating		IP62		IP31		
Power consumption		50 W		370 W		

Ilustración 6 Especificaciones técnicas del catálogo de escáneres InnovaSCAN

El hardware viene acompañado de un software propio que controla tanto la generación y proyección de los patrones como la captura de la nube de puntos en el espacio 3D.

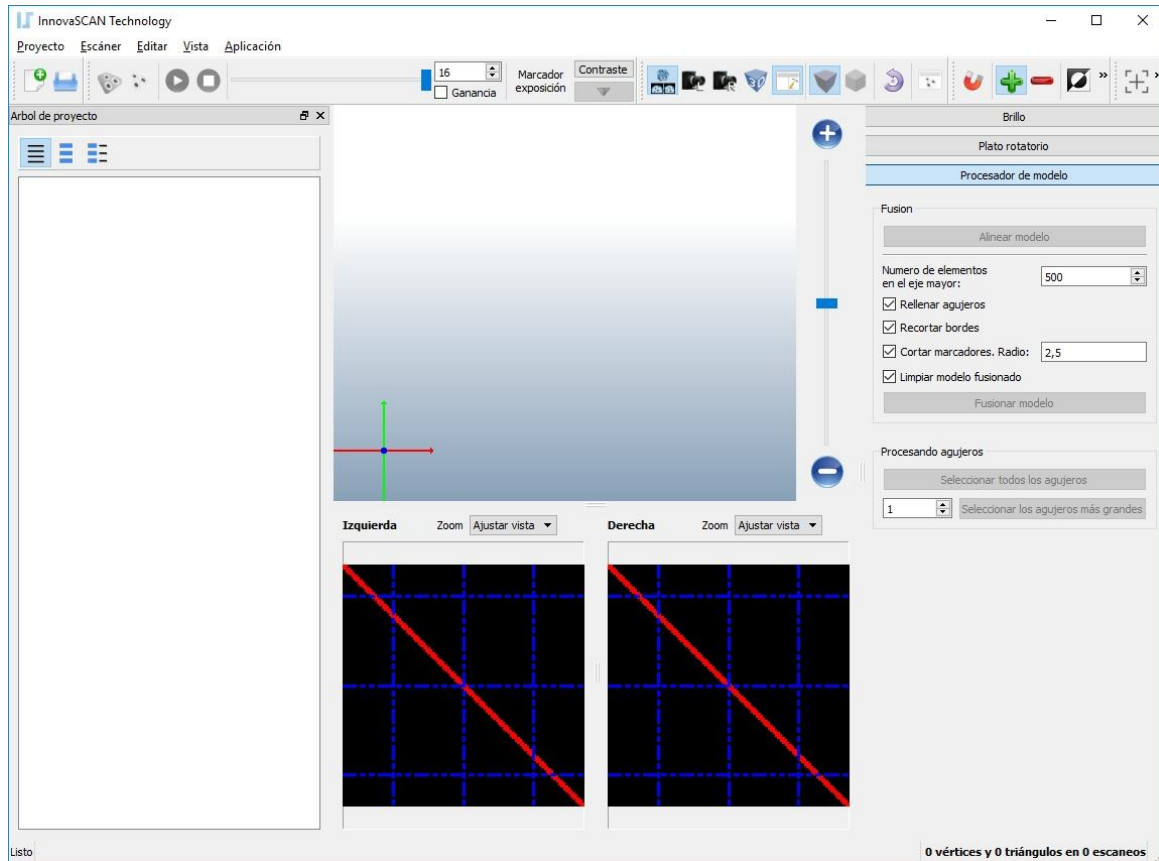


Ilustración 7 Software InnovaSCAN para el escáner de luz estructurada

A partir de los datos capturados desde varios ángulos del objeto el software recompone el volumen. Este método es propenso a errores y defectos debido a brillos producidos por el material y la luz ambiental, por lo que tener unas condiciones óptimas es difícil en la mayoría de los casos.

Para eliminar estos errores, los modelos obtenidos deben ser tratados con software de modelado y creación de geometría 3D. En el mercado existen infinidad de herramientas que se ajustan estas necesidades. Algunos de los más usados son [AutoDesk 3D Max](#), [Cinema 4D](#) o [Blender](#). Éste último es el usado actualmente en la empresa ya que es libre y brinda funcionalidades de simplificación y retopología de mallas muy potentes y fáciles de usar.



Blender es un programa multiplataforma, dedicado especialmente al modelado, iluminación, renderizado, animación y creación de gráficos en 3D. También de composición digital utilizando una técnica procesal de nodos, edición de vídeo, escultura (incluye topología dinámica) y pintura digital. En Blender, además, se pueden desarrollar videojuegos ya que posee un motor de juegos interno.

El programa fué inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre. Actualmente es compatible con todas las versiones de Windows, Mac OS X, GNU/Linux (Incluyendo Android), Solaris, FreeBSD e IRIX.

3.2. Creación del modelo con textura en alta definición

Para crear un modelo con una textura en alta definición usaremos la técnica de la fotogrametría. El modelo resultante no tendrá el nivel de densidad de puntos que el obtenido con el modelo anterior (de cerca se apreciarán formas más suavizadas) pero sí que se generará paralelamente una textura de alta calidad.

3.2.1. Fotogrametría: El método de reconstrucción

La fotogrametría es la técnica cuyo objeto es estudiar y definir con precisión la forma, dimensiones y posición en el espacio de un objeto cualquiera, utilizando esencialmente medidas hechas sobre una o varias fotografías de ese objeto (Schenk & Catalunya, 2002).

El método de reconstrucción de objetos o terreno (cartografía) mediante fotogrametría consiste de forma general en:

- Fotografiar los objetos: Se deben hacer la suficiente cantidad de fotos desde todos los ángulos posibles del objeto para que el procesado posterior pueda encontrar puntos comunes y proceder a la construcción del volumen 3D. Esta fotos deben tener una calidad aceptable.
 - Hay que tener cuidado con la luz ambiental, dependiendo del material del objeto, los reflejos o zonas más oscuras darán bastantes problemas. Se ha comprobado que

materiales como el bronce en ambientes exteriores, a plena luz del día generan fotografías muy oscuras y con reflejo.

- Una posible solución sería transportar el objeto a una sala con luz controlada, como un estudio fotográfico si es posible. Otra solución no invasiva es ir rectificando el obturador de la cámara para intentar obtener las mejores fotos posibles.
- Orientación de las imágenes: Colocación de las fotografías en la posición adecuada con sus marcas de referencia (orientación interna). Colocar los fotogramas en la misma posición que ocupaban entre ellos en el momento de las tomas (orientación relativa).
 - Formación del modelo por restitución para después aplicarle giros, una traslación y un factor de escala (orientación absoluta) para tener el modelo (objeto) en coordenadas terreno. Incluye también el escalado del objeto para obtener y realizar medidas en las magnitudes reales.
 - Formación del modelo por rectificación, consistente en, una vez aplicados la orientación tanto interna como externa del haz de luz, hallar la intersección entre dicho haz orientado y el modelo digital del terreno correspondiente al espacio que se quiere determinar. Para realizar una rectificación se ha tenido que realizar previamente una restitución de dicho lugar (Schenk & Catalunya, 2002).



Ilustración 8 Modelado de un busto usando fotogrametría con Agisoft PhotoScan, Nick Lievendag

3.2.2. Equipo para fotogrametría

Para crear un modelo mediante fotogrametría necesitamos 2 componentes indispensables. El primero consiste en un cámara reflex capaz de tomar fotos alta resolución. El segundo es algún programa que tome esas fotos como entrada y realice el proceso fotogramétrico en su totalidad. Como se ha mencionado anteriormente la calidad y cantidad de éstas fotografías determinarán el resultado final y el tiempo que necesitará el software para reconstruir el modelo.

Para la toma de fotos, la empresa dispone de una cámara reflex Sony DSLR-A-380 con un objetivo de 18-55mm.



Ilustración 9 Cámara Sony DSLR-A-380 usada para fotogrametría

Para la reconstrucción del modelo se han usado varios productos software desde el inicio de esta línea de trabajo. Actualmente se usa un programa llamado [Context Capture](#), desarrollado por Bentley Systems, por la calidad de sus resultados y un tiempo bajo de procesamiento con respecto a otras soluciones existentes en el mercado.

3.3. Modelado: “La Inmaculada Napolitana”



Ilustración 10 Inmaculada Napolitana después del proceso de restauración 1

Ficha Técnica	
Nombre	Inmaculada
Origen	Escuela napolitana
Época	Entre 1690 y 1720
Perteneciente a	Retablo-Relicario de la Sacristía Mayor. Catedral de Jaén
Dimensiones	97 x 49 x 39 cm

La empresa ha tenido el privilegio de poder escanear esta obra de arte antes de su restauración, gracias al excelentísimo Cabildo de la Catedral de Jaén y la Universidad de Jaén.



Ilustración 11 Inmaculada Napolitana después del proceso de restauración 2

Siguiendo los procesos anteriormente descritos se han obtenido, mediante escáner de luz estructurada un modelo en alta resolución de la talla sin color y, mediante fotogrametría, otro modelo con color (textura de alta definición).



Ilustración 12 Proceso Escáner de luz estructurada 1

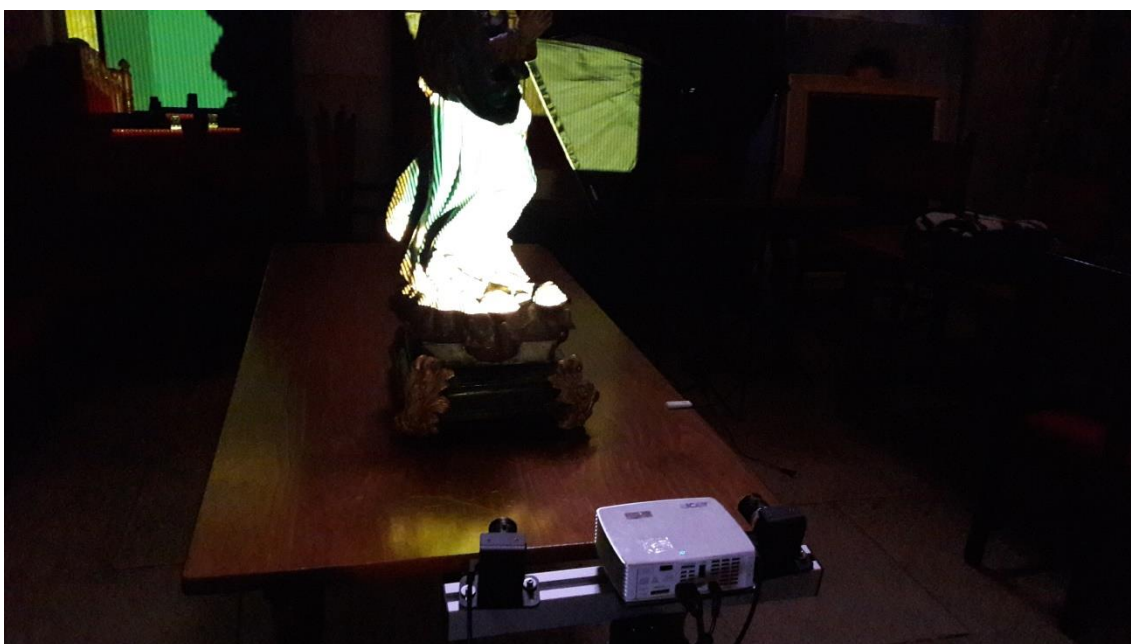


Ilustración 13 Proceso Escáner de luz estructurada 2

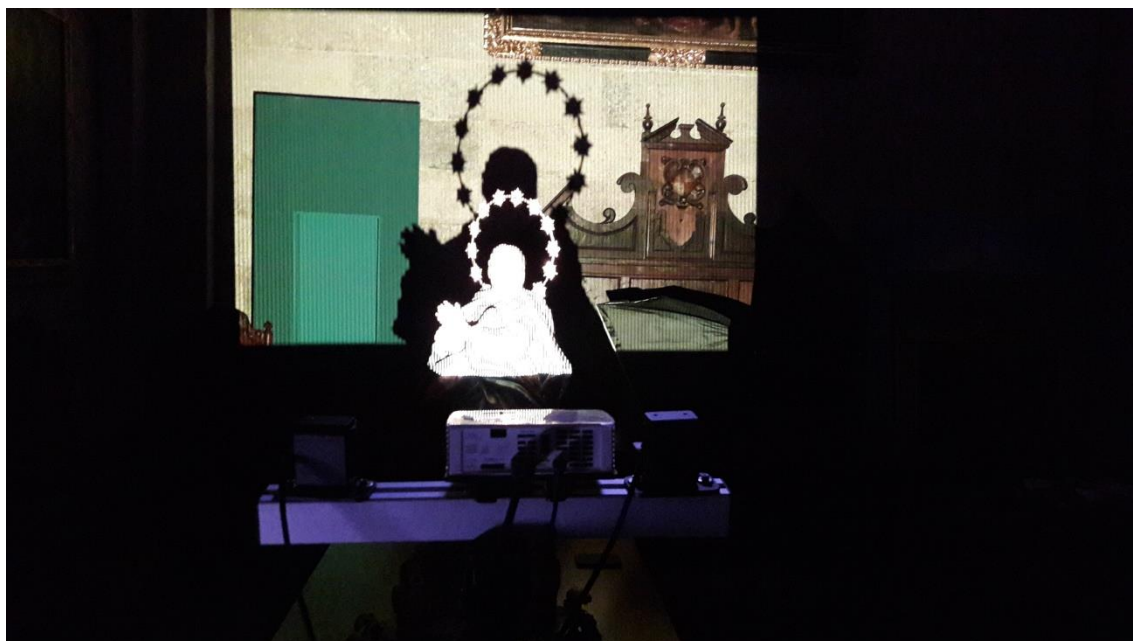


Ilustración 14 Proceso Escáner de luz estructurada 3



Ilustración 15 Proceso Fotogrametría 1



Ilustración 16 Proceso Fotogrametría 2



Ilustración 17 Proceso Fotogrametría 3

3.3.1. Modelos Obtenidos

A continuación se presentan los resultados obtenidos mediante el uso de las técnicas de escaneado mediante escáner de luz estructurada y fotogrametría.



Ilustración 18 Modelo a máxima resolución

La *Ilustración 18* muestra una sección del modelo 3D obtenida a la máxima resolución con el escáner de luz estructurada. Puede apreciarse el más mínimo detalle o imperfección sobre la superficie de la obra.

Este modelo tiene unas dimensiones que pueden ser admisibles para renderización (archivo .obj de 2 Gigabytes), pero para cargarlo en tiempo real y manipularlo no es óptimo y generará problemas de memoria y largos tiempos de espera. Para la exposición del futuro proyecto a desarrollar se usará una versión simplificada del modelo, con 1.300.000 puntos y 2.600.000 triángulos cuyo .obj ocupa 300 Megabytes.



Ilustración 19 Modelo en media resolución

Seguidamente se mostrarán pares de imágenes, a la derecha la fotografía real y a la izquierda el modelo generado mediante escáner de luz estructurada con la textura generada mediante fotogrametría.



Ilustración 20 Realidad frente a Modelo 1



Ilustración 21 Realidad frente a Modelo 2

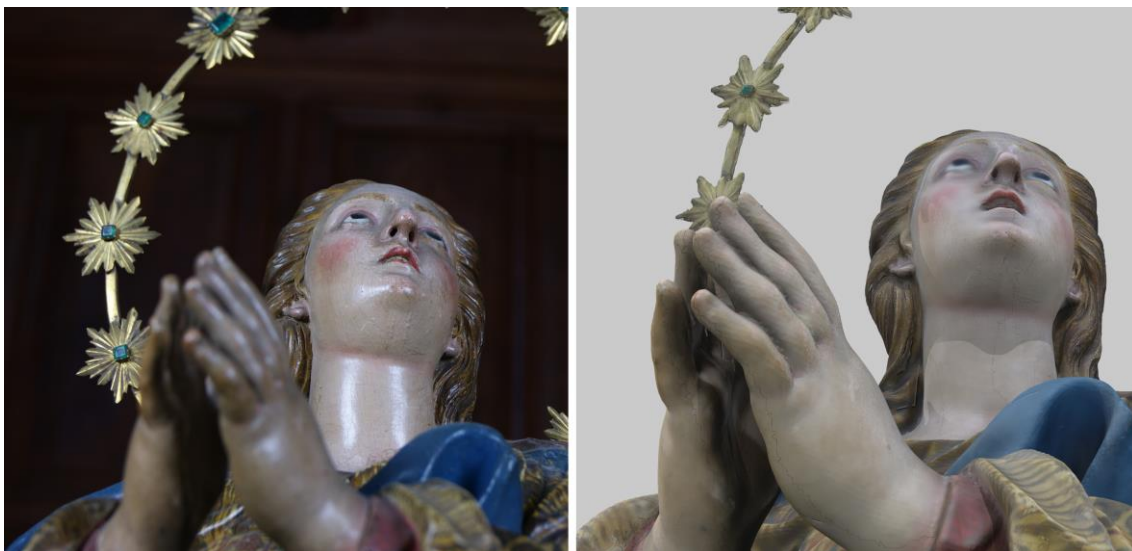


Ilustración 22 Realidad frente a Modelo 3



Ilustración 23 Realidad frente a Modelo 4



Ilustración 24 Realidad frente a Modelo 5



Ilustración 25 Realidad frente a Modelo 6



Ilustración 26 Realidad frente a Modelo 7

UNIDAD II: EL PROCESO DE DESARROLLO

“Hay dos maneras de diseñar software: una es hacerlo tan simple que sea obvia su falta de deficiencias, y la otra es hacerlo tan complejo que no haya deficiencias obvias. El primer método es mucho más difícil.”

Charles Antony Richard Hoare

4. Análisis de requisitos

Para obtener los principales requisitos del sistema a desarrollar se debe definir junto con el cliente el alcance del proyecto y acto seguido crear la estructura de desglose de requisitos (RBS), la cual servirá de referencia a clientes y gestores de proyectos.

4.1. Alcance del proyecto

Tras varias reuniones con el equipo al completo de la empresa, se ha redactado una descripción del sistema que posteriormente servirá como base para obtener una colección de los principales requisitos del sistema, definiendo así el alcance del proyecto. El desarrollo priorizará la parte relacionada con la difusión de patrimonio debido a la posibilidad de aprovechar una futura exposición que dará visibilidad al producto y a la empresa ([11 Exposiciones](#)). A continuación se presenta la descripción del sistema:

Partiendo de los modelos 3D obtenidos por la empresa en su línea de trabajo [1.3.3 Tecnologías 3D para la Conservación y Difusión del Patrimonio](#), se requiere el desarrollo interno de una aplicación interactiva que presente los modelos 3D con su texturizado (e información asociada a éstos) a usuarios visitantes en galerías y museos. El sistema debe permitir definir áreas sobre los modelos (puntos de interés). Se podrán asociar textos, imágenes, audios y videos a estas áreas que posteriormente serán presentadas a los usuarios. La carga y guardado de los modelos 3D y texturas debe realizarse en tiempo de ejecución, siendo esta espera tolerable para el usuario.

El sistema debe organizar estos modelos 3D, sus texturas e información por proyectos. Un proyecto puede contener varios modelos y texturas a la vez. Pueden existir submodelos que muestren con más detalle algunas porciones del modelo base o que posean información adicional como texturas en ultravioleta o infrarrojos. Estos modelos secundarios no contendrán puntos de interés. La creación de estos proyectos será una funcionalidad privada interna de la empresa. Ningún usuario podrá crear proyectos o añadir modelos a un proyecto existente.

Se debe poder elegir la plataforma objetivo, pudiendo compilar el programa para Windows o Android con poco esfuerzo, los sistemas operativos más comunes en tablets y pantallas táctiles. Los proyectos serán portables entre las diferentes versiones (Windows y Android). En la versión Android no serán necesarios los mecanismos para incluir elementos multimedia. Debe permitir una interacción táctil por parte de los usuarios, rotando los modelos 3D y navegando por la aplicación. Esta interacción se podrá realizar por medio del ratón, teclado o deslizando los dedos sobre una pantalla táctil.

El sistema debe ofrecer varios idiomas a los usuarios. Un idioma nueva ha de poder incluirse fácilmente en la aplicación. La escena principal debe poder customizarse, empezando por ajustar los niveles de luz que afectarán a los modelos y el color de fondo.

4.2. Principales requisitos del proyecto

Esta es la lista de los principales requisitos extraídos de la anterior definición, priorizados por el equipo de la empresa. Con esta información podremos construir la estructura de desglose de requisitos (RBS) que servirá como base para obtener la estructura de desglose del trabajo (WBS).

Requisitos Principales
Guardado/Cargado de modelos 3D y texturas en tiempo real.
Interacción por parte de los usuarios mediante ratón, teclado y eventos táctiles en pantallas que lo permitan. El usuario debe poder rotar y hacer zoom en el modelo mediante estas entradas.
Sistema de proyectos para organizar la información. Un proyecto debe poder contener varios modelos y texturas asociadas. La creación de proyectos será una funcionalidad interna de la empresa, al igual que la inclusión de nuevos modelos 3D en un proyecto. Los proyectos serán portables entre las diferentes versiones (Windows y Android)
Windows y Android deben poder definirse como plataformas objetivo. El sistema debe poder ejecutarse con la misma funcionalidad (salvo creación de proyectos) en ambos entornos.
Sistema de puntos de interés sobre la geometría para asociar información multimedia. Se deben poder definir áreas que contengan información (texto, imágenes, audio y videos). Esta información será ofrecida a los usuarios cuando consulten los puntos de interés. En la versión Android no serán necesarios los mecanismos para incluir elementos multimedia. Los modelos secundarios no contendrán puntos de interés
Internacionalización del sistema, con fácil inclusión de nuevos idiomas. El sistema de ofrecer una lista de idiomas disponibles para su interfaz. Estos idiomas deben aplicarse en tiempo real, sin cerrar o reiniciar la aplicación. Incluir un fichero debe ser una tarea simple y fácil.
El sistema debe permitir personalizar la escena ajustando la intensidad de luz y cambiando el color de fondo.

Tabla 1 Requisitos principales extraídos de la descripción del sistema

4.3. Estructura de Desglose de Requisitos

El nivel superior del RBS (Requirement Breakdown Structure) está constituido por los principales requisitos. Una de las ventajas del RBS es que es una representación intuitiva y cercana al cliente. Muestra una visión clara del grado en que la solución está definida. Permite elegir el modelo de ciclo de vida de la gestión del proyecto por su grado de completitud. Constituye la parte superior del WBS (Work Breakdown Structure), el cual será construido en la fase de planificación (6 *Planificación*). Este documento sirve de referencia al cliente desde la fase de definición del ámbito del proyecto. En este caso concreto se ha usado la herramienta [Xmind](#) para la generación del siguiente diagrama, *Ilustración 27*.

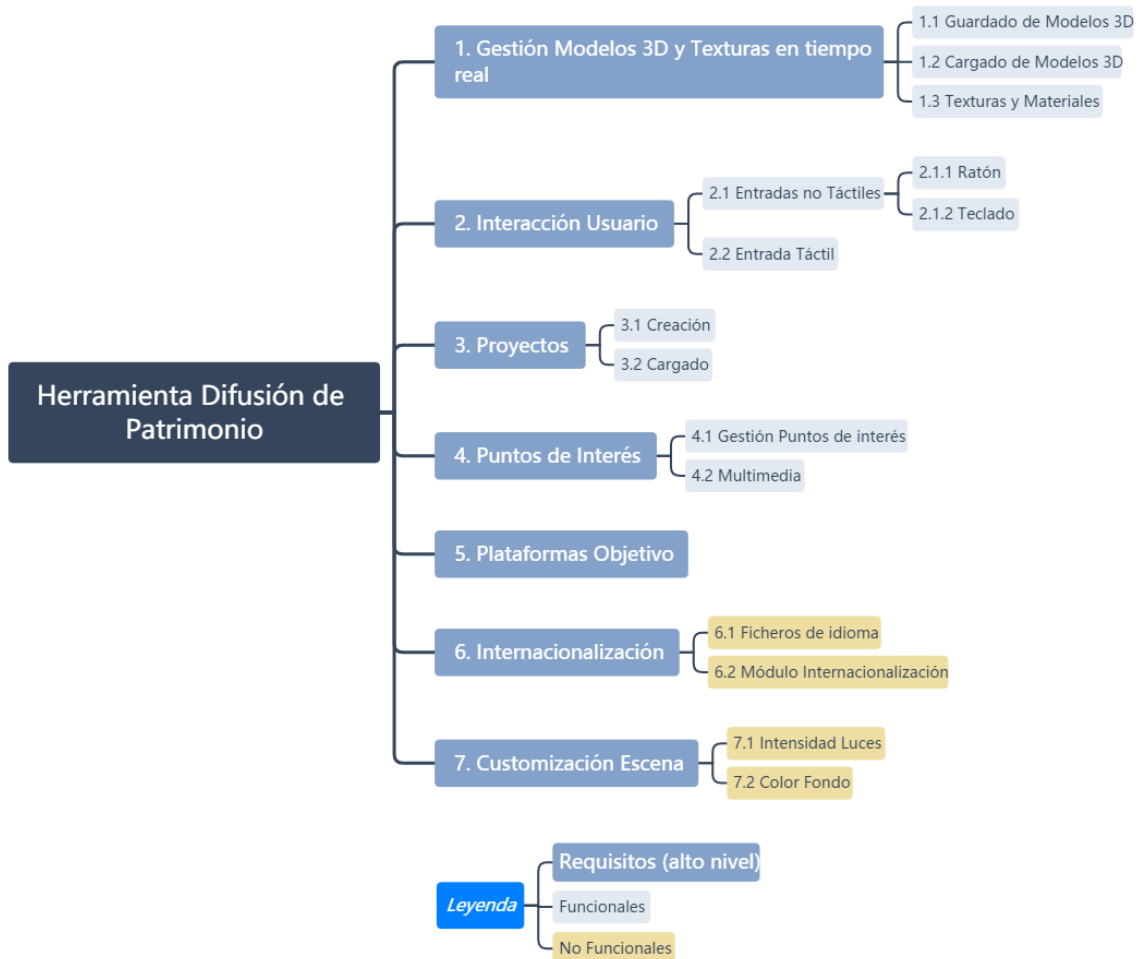


Ilustración 27 Diagrama RBS del proyecto

4.4. Ciclo de Vida de la Gestión del Proyecto

Para la planificación y ejecución de las futuras tareas de desarrollo que se definirán en la etapa de planificación se ha optado por el uso de una metodología ágil (Scrum). La razón que lleva a esta decisión es que el equipo de desarrollo no tiene demasiada experiencia en algunas partes esenciales del proyecto que podrían requerir más horas de dedicación.

4.4.1. ¿Por qué Scrum?

Scrum es un modelo de desarrollo ágil caracterizado por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra en un ciclo secuencial o de cascada (Martel, 2014).

Teniendo en cuenta la naturaleza del proyecto, una metodología ágil es totalmente aplicable. Se realizará un aprendizaje por descubrimiento, una característica propia de metodologías ágiles. Se trata de un proyecto de pequeño tamaño, susceptible a modificaciones durante el desarrollo, del que no depende la economía de la empresa, por lo que resulta un escenario perfecto para usar y ganar experiencia con esta metodología.

4.4.2. Adaptando Scrum

Aun siendo un solo desarrollador, el equipo de la empresa al completo seguirá el proceso de desarrollo muy de cerca y estarán presentes tanto en las reuniones de planificación como en las de retroalimentación, siendo estas últimas las más importantes. El equipo aportará ideas en la fase de desarrollo y probarán el producto minuciosamente en las fases de pruebas. Se utilizará una futura exposición para testear el producto con usuarios reales. De esta forma se adapta la metodología a este caso concreto, aumentando las probabilidades de éxito.

5. Elección de Tecnologías y Herramientas

Partiendo de los requisitos obtenidos en el punto anterior se extraen las características deseables a las que se debe prestar atención para elegir la tecnología a emplear en este proyecto. Algunos de los puntos más importantes son:

- Tecnología asequible económicamente o libre
- Curva de aprendizaje poco pronunciada
- Tecnología actual, con actualizaciones frecuentes
- Multiplataforma (Windows y Android)
- Cargado/guardado eficiente de modelos 3D, texturas y otros elementos multimedia en tiempo real.
- Fácil uso y programación de las entradas de usuario (control táctil)
- Cómoda creación y uso de shaders y materiales
- Fácil manipulación de los elementos típicos de una escena 3D (luces, físicas, modelos primitivos,...)
- Gestión de archivos del sistema

Al tratarse de un proyecto con una moderada interacción con el usuario y un gran peso gráfico y multimedia los motores de videojuego o los motores de renderizado son candidatos muy viables para este proyecto, aportando muchas de las funcionalidades que serán necesarias o simplificando la implementación de éstas.

Como base para el desarrollo del sistema se han tenido en cuenta los 3 motores de videojuego más famosos y extendidos del momento: Unreal Engine, Unity 3D y CryEngine, y un motor de renderizado llamado Ogre 3D.

5.1. Motores de Videojuego

Se define como motor gráfico al framework de software diseñado para crear y desarrollar videojuegos. Los desarrolladores pueden crear videojuegos para consolas, dispositivos móviles u ordenadores usando este software. La funcionalidad básica de un motor es proveer al videojuego de un motor de renderizado para los gráficos 2D y 3D, motor físico o detector de colisiones, sonidos, scripting, animación, inteligencia artificial, redes, streaming, administración de memoria y un escenario gráfico.

El uso de un motor de videojuego en el proyecto proporciona herramientas de alto nivel para la manipulación en tiempo real de modelos 3D y sus texturas. Cabe destacar que la mayoría de estos motores añaden funcionalidades extra que podrían ser usadas en el proyecto, acelerando el tiempo de desarrollo. Estas características serían, la creación de menús interactivos, manejo de eventos o una gestión de las entradas de usuario

uniforme e independiente a las plataformas objetivo para las que se compilará el proyecto, entre otras.

5.1.1. Unreal Engine



UE4 tiene algunas capacidades gráficas increíbles, incluyendo características como capacidades avanzadas de iluminación dinámica y un nuevo sistema de partículas que puede manejar hasta un millón de partículas en una escena a la vez. Unreal Engine ofrece varias herramientas adicionales de gran ayuda para diseñadores y artistas. La facilidad de uso del UE4 hace que sea mucho más atractivo para los nuevos desarrolladores de juegos.

El motor Unreal fue originalmente desarrollado principalmente para los FPS (First Person Shooter), y el primer juego que fue diseñado como demo fue el aclamado Unreal, pero desde entonces ha sido utilizado para una gran variedad de juegos y géneros como los juegos de rol.

El lenguaje usado para scripting es C++, un lenguaje muy familiar para la mayoría de los desarrolladores.

Ofrece un sistema llamado Blueprint, el cual puede usarse para “programar” sin tener que escribir una sola línea de código. Los blueprints y el código C++ están pensados para complementarse. Todos los blueprints pueden exportarse y se transformarán en código C++ que podremos modificar carácter a carácter.

Cada mes, Epic Games lanza nuevas versiones de UE4 con suculentas mejoras. A nivel gráfico y estético el motor es inmejorable, y valga la redundancia, además mejora mes a mes. La gestión de partículas es la mejor del momento, ahora hacer superficies que reflejen en tiempo real no supone toparse con múltiples dificultades, gracias al nuevo sistema de raycasting, la gestión de transparencias ya no tiene un orden arbitrario.

UE4 viene con su propio gestor de menús 2D y 3D. Cuenta con un módulo (Papers 2d) increíblemente potente para gestionar juegos en 2D dimensiones de cualquier tipo: profundidad, colisiones 2D, físicas restringidas en Z, gestión de orden de elementos y menús.

Está más orientado a equipos grandes y el desarrollo de juegos de gran calidad (triple A). Su motor gráfico es el más potente de la comparativa (). Al igual que Unity, está basado en componentes.

Cabe señalar que Unreal Engine 4 está disponible gratuitamente en su totalidad, pero a partir de obtener unas ganancias de 3000\$ con un producto desarrollado sobre este motor se deberá pagar el 5% de los beneficios brutos posteriores (royalties).

Unreal Engine ha sabido adaptarse a las demandas del sector mediante una estrategia agresiva y arriesgada de desarrollo abierto y esto ha hecho que el motor coja una carrerilla y una ventaja respecto a sus competidores que ahora lo tienen más difícil que nunca.

5.1.2. Unity 3D



Unity 3D ofrece una amplia gama de características y su interfaz es bastante fácil de entender. Se caracteriza por su gran integración multiplataforma, es decir, los juegos se pueden trasladar de forma rápida y fácilmente en Android, iOS, Windows Phone 8 y BlackBerry, por lo que es un gran motor de juego para el desarrollo de juegos para móviles. También tiene la capacidad de desarrollo para Playstation 3, Xbox 360, Wii U y navegadores web.

El motor de juego es compatible con las principales aplicaciones 3D como 3ds Max, Maya, Softimage, Cinema 4D, Blender y más, lo que significa que no hay restricciones reales al tipo de formatos de archivo que soporta. Implementa Sprites de apoyo y la física en 2D, por lo que es un gran motor de juego que se utiliza para el desarrollo de juegos 2D.

Unity no posee características reales de modelado o construcción más allá de unas pocas formas primitivas, de manera que los equipos de desarrollo deberán apoyarse en aplicaciones de terceros. Sin embargo, cuenta con una tienda de Assets (Asset Store) desde la que se puede descargar y usar infinidad de recursos gratuitamente o por poco dinero (Subero, 2014).

Actualmente la versión Personal de Unity 3D es gratuita con todas sus características disponibles excepto desarrollo para consolas y otros servicios adicionales como Unity Cloud Build (herramienta para la integración automática

de cambios). La pega es que si se compila un juego en esta versión tendremos una pantalla al inicio de la ejecución del videojuego con el logo de Unity que no se podrá eliminar. Se puede usar la versión Personal para producción si los ingresos brutos al año son menores de 100.000\$.

Existe una versión Profesional (desde 75\$ al mes) la cual posee todas las características y servicios disponibles.

Unity 3D Engine es perfecto para equipos pequeños y el desarrollo para plataformas móviles. La curva de aprendizaje de Unity es la más fácil de los tres motores pero su motor gráfico es el menos potente. Es un motor de juegos de propósito general.

Ventajas

- Posee una amplia documentación y multitud de videos explicativos. Al llevar más tiempo accesible económicamente a muchos usuarios, la cantidad de información sobre scripting, plugins desarrollados por terceros y recursos es enorme.
- A la hora de implementar los Scripts, se puede elegir entre dos lenguajes de programación: C# y JavaScript. Anteriormente también se podía programar en Boo.
- Está basado en componentes, lo que invita a la reutilización y estructuración de las entidades usadas.
- Los requisitos técnicos para trabajar con este motor son medios-bajos.
- La Asset Store ofrece infinidad de soluciones ya creadas, modelos, animaciones,... Dispone de muchos componentes gratuitos.
- Es ideal para la creación prototipos.
- El IDE que nos ofrece por defecto (Monodevelop) es muy fácil de usar. Permite utilizar otros como Visual Studio con mucha facilidad.

Desventajas

- No está muy orientado a ser usado por diseñadores o modeladores.
- Su versatilidad para el desarrollo en diferentes plataformas es activada con licencias de pago que pueden elevar bastante el coste de producción.

5.1.3. CryEngine



CryEngine es un motor muy potente diseñado por la empresa de desarrollo de Crytek que se introdujo en el primer juego de la saga Far Cry. Está diseñado para ser utilizado en plataformas móviles, PC y consolas, incluyendo PlayStation 4 y Xbox One.

Las capacidades gráficas de CryEngine están a la par con el Unreal Engine 4, con una iluminación fantástica, la física realista, sistemas avanzados de animación y mucho más. Similar a UE4, CryEngine tiene características intuitivas y potentes de diseño de niveles en el motor del juego.

CryEngine es un motor de juego muy potente, no obstante su curva de aprendizaje es un poco complicada. Para empezar a utilizar el motor de juego de manera productiva, y puede ser más difícil de entender si no se tiene ninguna experiencia con motores de juegos.

Es especialmente sencilla la creación de escenarios exteriores con terrenos de gran amplitud. La documentación existente es extensa y hay foros oficiales para desarrolladores. Está diseñado para que equipos multidisciplinares trabajen conjuntamente sobre un mismo proyecto.

Actualmente sigue un modelo de negocio parecido al de los dos motores anteriores, en el que después de ganar 5.000 dólares con un producto desarrollado con esta tecnología se deberá pagar un 5% de de los beneficios brutos posteriores.

5.2. Motores de Renderizado

Debido a las ventajas que puede ofrecer un motor de renderizado, tales como un extenso conjunto de funciones de alto nivel para el tratamiento de gráficos acelerados por hardware, se estudió la viabilidad del uso de uno de ellos en el proyecto.

5.2.1. Ogre 3D



Ogre 3D (Object-Oriented Graphics Rendering Engine) es un motor de renderizado 3D orientado a escenas (en oposición a un motor de videojuego), escrito en el lenguaje de programación C++ y software libre. Fue diseñado para que a los desarrolladores les resulte más fácil e intuitiva la producción de aplicaciones que utilizan gráficos 3D acelerados por hardware. Sus bibliotecas evitan la dificultad de la utilización de capas inferiores de librerías gráficas como OpenGL y Direct3D, y además, proveen una interfaz basada en objetos del mundo y otras clases de alto nivel.

OGRE es "solo" un motor de renderizado. Como tal, su propósito principal es proveer una solución general para el renderizado de gráficos. A pesar de que incorpora otras características (clases de vector y matrices, manejo de memoria, etc.), son consideradas suplementarias. No es una solución todo-en-uno desde el punto de vista del desarrollo de juegos o simulación ya que no provee soporte para físicas y audio, por ejemplo (Ogre3D Team, 2016).

La elección de OGRE como un motor gráfico otorga a los desarrolladores la libertad de usar librerías de física, entrada, audio y otras que deseen y permite al equipo de desarrollo de OGRE enfocarse en la gráfica en lugar de distribuir sus recursos en varios sistemas.

5.3. Comparativa

A continuación se muestra la tabla resumen, [Tabla 2](#), que se ha creado para elegir la tecnología que más ventajas aporte al proyecto.

Comparativa Tecnologías	Unreal Engine 4	CryEngine 5	Unity 3D	Ogre 3D
Curva de Aprendizaje	Media	Difícil	Fácil	Media
Calidad Gráfica	Muy Alta	Alta	Media	Alta
Características de modelado 3D y construcción de escenas avanzada	Sí	Sí	No	No
Apto para equipos pequeños	No	No	Sí	Sí
Basado en componentes	Sí	No	Sí	No
Calidad del Motor de físicas	Muy Bueno	Excelente	Bueno	No tiene
Gratis al inicio del proyecto	Sí	Sí	Sí	Sí, siempre
Gran biblioteca de recursos propios	No	No	Sí	No
Comunidad extensa	Sí	No	Sí	No
Calidad de la documentación	Excelente	Buena	Excelente	Buena
Tutoriales de terceros	Sí	No	Sí, muchos	No muchos
Android y Windows como plataforma objetivo	Sí	Sí	Sí	Sí
Gestión de entradas de usuario	Sí	Sí	Sí	No
Interfaz para la creación de escenas	Sí	Sí	Sí	No
Gestión de audio, video e imágenes integrada	Sí	Sí	Sí	No
Última versión	4.15	5.3.0	5.5.2	1.9.0
Fecha última versión	15/02/17	15/12/16	24/02/17	22/11/13

Tabla 2 Comparativa Tecnologías (27/02/2017)

Tras diversas pruebas con Ogre 3D, finalmente se descartó por el hecho de no aportar ninguna funcionalidad adicional a la gestión de gráficos. Se estimó que el tiempo de desarrollo sería muy superior comparado con el uso de cualquier otra de las opciones contempladas al tener que implementar desde cero muchas de las partes necesarias, como las entradas de usuario y la navegación por menús.

Quedando claro que se usará un motor de videojuego, se analizarán más a fondo las características de las tres opciones restantes. Unreal Engine y CryEngine son motores orientados al desarrollo de juegos de alta calidad, con un acabado visual brillante. Ambos motores están diseñados para ser usados por grandes equipos de manera colaborativa y no disponen de una gran variedad de componentes gratuitos desarrollados por la comunidad (la mayoría son de pago, lo que supondría un incremento en el coste o en el tiempo de desarrollo). Las curvas de aprendizaje no son tan ligeras como la de Unity 3D.

Unity 3D es un motor de videojuego de propósito general, usado para la creación de prototipos por su sencillez y rapidez. Se adapta mejor a equipos de desarrollo pequeños y su calidad gráfica es lo suficientemente buena como para ser considerado una opción válida.

Si a este hecho se le suma una importante experiencia previa por parte del equipo de la empresa con esta tecnología y una enorme cantidad de componentes y scripts libres implementados por la comunidad al alcance de cualquiera se tiene un claro favorito. Se usará Unity 3D para desarrollar ambas versiones de la herramienta de Difusión y Conservación de Patrimonio, Windows y Android.

6. Planificación

La fase de planificación de un proyecto software tiene un carácter crítico. Una buena planificación mejorará la eficiencia con la que una empresa gestiona sus recursos e incrementará la comprensión del proyecto por parte del equipo. Una mala o nula planificación puede incluso hacer que un proyecto muy prometedor tenga que abandonarse o que éste comprometa económicamente a la empresa hasta el punto de hundirla (Cardozzo & Academy, 2014).

Las pequeñas dimensiones del proyecto de este trabajo fin de máster, junto con su carácter lineal, no lo exigen de una mínima planificación. En anteriores proyectos de mayor envergadura se han usado herramientas de gestión de proyectos como Redmine. En este caso concreto, se ha descartado esta opción por diversos motivos:

- El tiempo semanal que el desarrollador puede dedicar al proyecto no es constante. Este desarrollo se ejecutará de forma paralela con otros proyectos y la realización de otras funciones en la empresa.
- La linealidad del proyecto antes descrita.
- El equipo desarrollador sólo consta de una persona.

No obstante se usará un sistema de control de versiones (GIT) para obtener un histórico detallado de los cambios producidos en el proyecto a lo largo del tiempo.

En primer lugar se deben conocer los recursos (materiales, humanos y económicos) que serán necesarios para afrontar este desarrollo. Partiendo del RBS obtenido en la fase de Análisis de requisitos (*4 Análisis de requisitos*), se deben descomponer los nodos hoja en tareas estimables en tiempo y esfuerzo. Estas tareas no deben estar presentes en más de una iteración, por lo que su tiempo debe ser menor al de ésta. Este proceso dará como resultado el WBS (Work Breakdown Structure) o estructura del desglose del trabajo. Esta estructura permitirá conocer una estimación de la duración de cada tarea y, por consiguiente, una estimación temprana del proceso de desarrollo al completo.

6.1. Estructura de Desglose del Trabajo

Como se ha mencionado anteriormente, el WBS es una extensión del RBS. Para que resulte fácilmente legible se ha cambiado el formato por una tabla en lugar de un diagrama, ya que la cantidad de nodos ha crecido exponencialmente.

Seguidamente se muestra el WBS, *Tabla 3*. Se han añadido 2 grupos de tareas extra, una al principio y otra al final. Estas tareas engloban las acciones pertinentes de apertura y cierre de proyecto. Se estimarán los nodos hoja (tareas) atendiendo a la experiencia previa del desarrollador.

Herramienta Difusión de Patrimonio	
TAREAS	TIEMPO ESTIMADO
0. Gestión de la Arquitectura de la Aplicación	
0.1 Diseño	1
0.2 Construcción del prototipo	0,5
0.3 Construcción de la escena	0,5
1. Gestión Modelos 3D y Texturas en tiempo real	
1.1 Guardado de Modelos 3D	
1.1.1 Diseño	1
1.1.2 Implementación	4
1.1.3 Pruebas de Unidad	1
1.2 Cargado de Modelos 3D	
1.2.1 Diseño	1
1.2.2 Implementación	3
1.2.3 Pruebas de Unidad	1
1.3 Texturas y Materiales	
1.3.1 Diseño	1
1.3.2 Implementación	6
1.3.3 Pruebas de Unidad	1
1.4 Pruebas de Integridad	1
2. Interacción Usuario	
2.1 Entradas no Táctiles	
2.1.1 Ratón	
2.1.1.1 Diseño	0,5
2.1.1.2 Implementación	2,5
2.1.1.3 Pruebas de Unidad	0,5
2.1.2 Teclado	
2.1.2.1 Diseño	0,5
2.1.2.2 Implementación	1
2.1.2.3 Pruebas de Unidad	0,5
2.2 Entrada Táctil	
2.2.1 Diseño	0,5
2.2.2 Implementación	3
2.2.3 Pruebas de Unidad	0,5
2.3 Pruebas de Integridad	0,5
3. Proyectos	
3.1 Creación	
3.1.1 Diseño	2
3.1.2 Implementación	3
3.1.3 Pruebas de Unidad	1
3.2 Cargado	
3.2.1 Diseño	1

3.2.2 Implementación	1
3.2.3 Pruebas de Unidad	1
3.3 Pruebas de Integridad	2
4. Puntos de Interés	
4.1 Gestión Puntos de interés	
4.1.1 Diseño	2
4.1.2 Implementación	8
4.1.3 Pruebas de Unidad	2
4.2 Multimedia	
4.2.1 Diseño	4
4.2.2 Implementación	15
4.2.3 Pruebas de Unidad	3
4.3 Pruebas de Integridad	4
5. Plataformas Objetivo	
5.1 Diseño	2
5.2 Adaptación	10
5.3 Pruebas	3
6. Internacionalización	
6.1 Ficheros de idioma	
6.1.1 Diseño	1
6.1.2 Creación Ficheros	1
6.2 Módulo Internacionalización	
6.2.1 Diseño	1
6.2.2 Implementación	6
6.2.3 Pruebas de Unidad	2
7. Customización Escena	
7.1 Intensidad Luces	
7.1.1 Diseño	1
7.1.2 Implementación	3
7.1.3 Pruebas de Unidad	1
7.2 Color Fondo	
7.2.1 Diseño	2
7.2.2 Implementación	2
7.2.3 Pruebas de Unidad	1
7.3 Pruebas de Integridad	1
8. Cierre del proyecto	
8.1 Generación de los ejecutables	1

Tabla 3 WBS del proyecto

El tiempo medio semanal dedicado a este proyecto será de 15 horas. Por lo que una iteración de 3 semanas equivale a 45 horas. Si se agrupan las horas de cada tarea por grupo de tareas al que pertenece se obtendrá la siguiente tabla, *Tabla 4*. En este caso los grupos coinciden con los requisitos

puesto que no existe ninguno que sobrepase la duración de la iteración y se haya tenido que desglosar en varios más pequeños.

Estimación Resumida por Grupos de Tareas	
Grupo de Tareas	Tiempo Estimado (Horas)
0.Gestión de la Arquitectura de la Aplicación	2
1.Gestión Modelos 3D y Texturas en tiempo real	20
2.Interacción Usuario	10
3.Proyectos	11
4.Puntos de Interés	38
5.Plataformas Objetivo	15
6.Internacionalización	11
7.Customización Escena	11
8.Cierre del proyecto	1
TOTAL:	119

Tabla 4 Estimación Temporal por grupos de tareas

6.2. Estimación de la Duración

Al configurar la iteración con una duración de 3 semanas y un tiempo de dedicación medio de 45 horas por iteración, se debe ajustar cuántas iteraciones existirán y qué tareas contendrá cada una de ellas. La **fecha de comienzo** prevista para el proyecto es el **06/03/2017**.

ITERACIONES			
Iteraciones	Grupos de tareas que comprende	Fecha Inicio	Duración Estimada (Horas)
Iteración 1	0,1,2,3	06/03/2017	43
Iteración 2	4	27/03/2017	38
Iteración 3	5,6,7,8	17/04/2017	38
TOTAL:			119

Tabla 5 Iteraciones

El proyecto tendrá una duración de 119 horas, repartidas en 3 iteraciones de 3 semanas cada una. **La fecha de finalización prevista es el 05/05/2017**.

6.3. Estimación de Recursos

Los recursos son las piezas esenciales que impulsarán y harán realidad el proyecto. En cualquier empresa el recurso indispensable y más importante son las personas. Además, la empresa debe disponer de bienes o activos que faciliten la correcta realización del trabajo u ofrezcan formas efectivas en dinero y tiempo de llevar a cabo las tareas requeridas .

6.3.1. Personas

Ya que el proyecto será realizado por un solo desarrollador, basta con calcular cuánto le cuesta a la empresa esas 114 horas de desarrollo. A fecha de marzo de 2017 se tiene el primer contrato en prácticas con la empresa, con una hora de trabajo valorada en 6,36 € brutos. El coste humano estimado del proyecto es de **725,04€**.

6.3.2. Recursos Técnicos

Aquí se recogen las infraestructuras y el software necesarios para la realización de proyecto.

6.3.2.1. Infraestructuras

El portátil usado por el desarrollador es propio, por lo que se computará una cuota de uso. Se va a suponer que este activo tendrá un valor residual igual a cero y que su vida útil es de 5 años. Una hora de uso de este dispositivo está valorada en 0,06€ por lo que su coste en el proyecto es de 7,80€.

Se usará para pruebas una tablet Samsung Galaxy Tab A (2016) que la empresa ganó en un concurso, valorada en 250 €.

Para la exhibición en las exposiciones la Universidad de Jaén ha adquirido y ha prestado a la empresa un ordenador All in One 23,8" HP 24-G013NS (2017) con pantalla táctil de 23'8 pulgadas, valorado en 800€.

Infraestructuras Requeridas			
Concepto	Precio de Adquisición	Cuota de uso (por hora)	Coste
Portátil MSI GE70	1.000,00 €	0,06 €	7,80 €
Tablet Samsung	800 €	0 €	0 €
Ordenador All in One	250 €	0 €	0 €
TOTAL:			7,80 €

Tabla 6 Infraestructuras requeridas para el proyecto

No se pueden computar servicios fijos tales como internet, luz, alquiler o agua, ya que la empresa disfruta de una oficina en el vivero del Antiguo Edificio de Magisterio de la universidad de Jaén, donde recibe estos suministros gratuitamente.

6.3.2.2. Software

Herramientas software indispensables para el desarrollo del proyecto.

Software Requerido		
Concepto	Razón	Precio
GIT	Control de versiones	0 €
Unity 3D 5.5.2	Framework para desarrollar el videojuego	0 €
Blender 2.78b	Retoques en los modelos 3D	0 €
TOTAL:		0 €

Tabla 7 Software requerido para el proyecto

6.4. Estimación de Costes

Con los datos calculados anteriormente se puede calcular la inversión total que necesitará el proyecto.

RESUMEN DE COSTES	
Recursos Humanos	725,04 €
Infraestructuras	7,80 €
Software	0 €
TOTAL:	732,84 €

Tabla 8 Resumen de Costes

El coste en infraestructuras es reducido ya que no se computa el equipo “prestado” o el que ya sea propiedad de la empresa. Los gastos de software son cero ya que se usan tecnologías libres u otras como Unity las cuales son gratis hasta traspasar cierto umbral de beneficios

7. Diseño Estructural

Antes de comenzar con el proceso de desarrollo se diseñará la estructura general de los componentes del sistema. Esta estructuración busca respetar el principio de responsabilidad única, manteniendo un bajo nivel de acoplamiento entre diferentes clases y una alta cohesión, principios básicos en los patrones de diseño GRASP (General Responsibility Assignment Software Patterns).

También se adjuntan el árbol jerárquico de la escena 3D dada su importancia y los patrones más relevantes que se usarán en el proyecto.

7.1. Diagrama de Componentes del Sistema

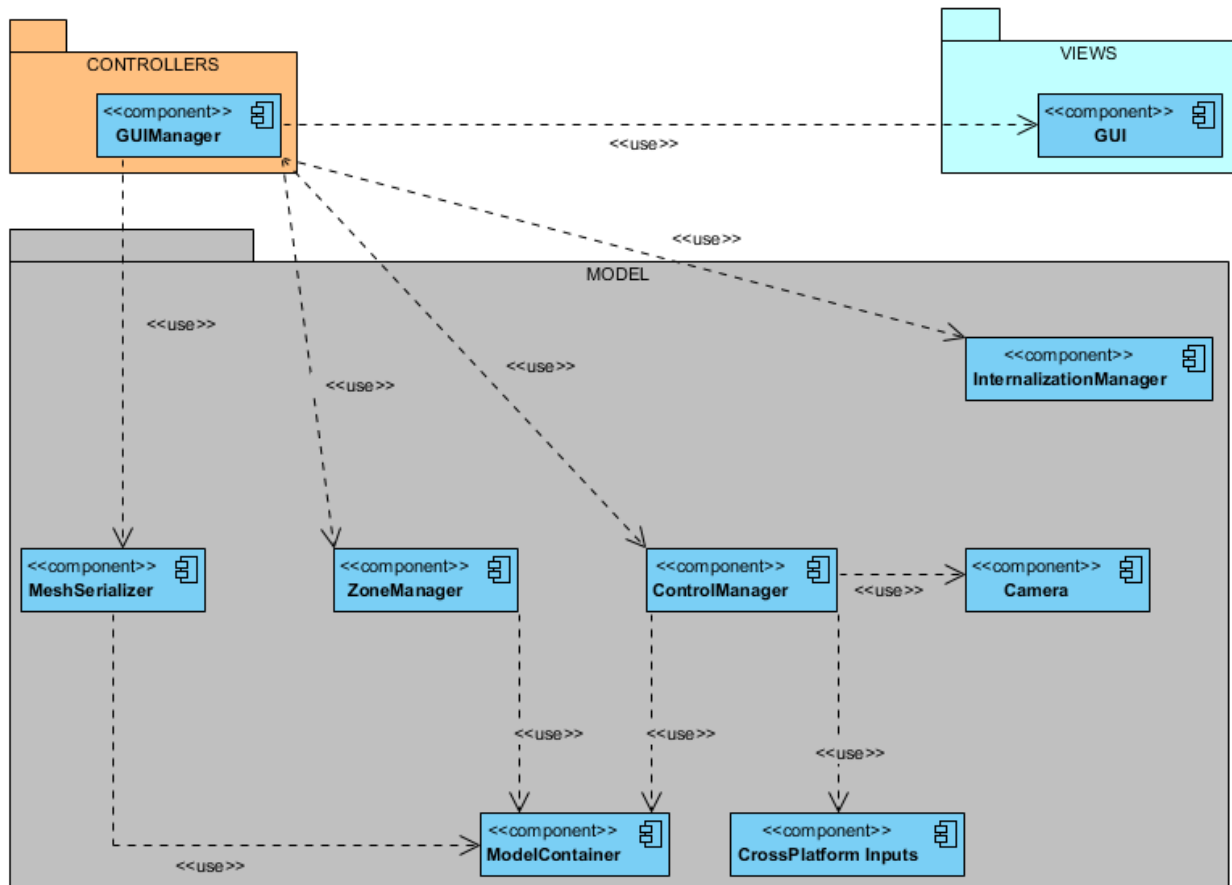


Ilustración 28 Diagrama de Componentes del sistema

Este diagrama, *Ilustración 28*, refleja las interdependencias y relaciones entre los diferentes módulos (o componentes) del sistema. Para clarificarlo se describirán los componentes más relevantes implicados en el funcionamiento.

7.1.1. GUI

Este componente representa la interfaz gráfica de usuario con los menús por los que éste podrá navegar. Este componente generará eventos sobre el entorno gráfico 3D de la aplicación (hacer click en un modelo, por ejemplo) que deberán ser capturados y derivados al GUI Manager. También presentará información (contenido multimedia) al usuario.

7.1.2. GUI Manager

Componente necesario para separar el sistema y las vistas. Su función es comunicar ambas partes y mantener cada una actualizada cuando se producen cambios o eventos en la otra. Se trata de implementar el patrón arquitectónico MVC (Modelo-Vista-Controlador).

7.1.3. Internalization Manager

Componente que traducirá los elementos de la vista (GUI) a un idioma en concreto. Ofrecerá una lista dinámica de los idiomas disponibles, cargará desde disco la información correspondiente y la aplicará a la interfaz.

7.1.4. Mesh Serializer

Módulo que gestionará los modelos 3D y sus texturas. Realizará el guardado (a la hora de crear los proyectos) y la correcta carga de estos elementos en tiempo de ejecución.

7.1.5. Zone Manager

Componente que gestionará los puntos de interés asociados a cada modelo del proyecto. Los puntos de interés y la información multimedia asociada a éstos será recuperada y modificada a través de este módulo. También mantendrá la representación gráfica de estos elementos en la escena.

7.1.6. Control Manager

Este componente recibirá eventos de las entradas de usuario (del componente CrossPlatform Inputs) y aplicará las siguientes transformaciones en los componentes:

- **Camera**: Funciones de Zoom y “Freeze” (efecto de congelación) de cámara.
- **Model Container**: Rotación sobre los 3 ejes del modelo cargado.

7.1.7. Camera

Este componente representa la cámara de la escena. Se pueden realizar las transformaciones básicas (traslación, rotación y escalado) sobre este elemento y otras modificaciones propias tales como aplicación de efectos visuales (antialiasing, ajuste de color,...), filtrado de elementos a renderizar por capas o definición de planos de corte entre otras muchas.

7.1.8. CrossPlatform Inputs

Este componente unificará las diferentes entradas de usuario (teclado, ratón y eventos de pantalla táctil) para simplificar el sistema.

7.1.9. Model Container

Componente situado en la coordenada origen (0,0,0) de la escena. Será un simple contenedor (objeto vacío) en el que se cargarán los modelos 3D. Tener un nodo padre del que cuelga todo el modelo 3D en su totalidad facilitará transformaciones como la rotación o el escalado y simplificará las tareas de limpieza de la escena 3D cuando un modelo ya no sea necesario.

7.2. Diagrama de Jerárquico de la Escena Principal

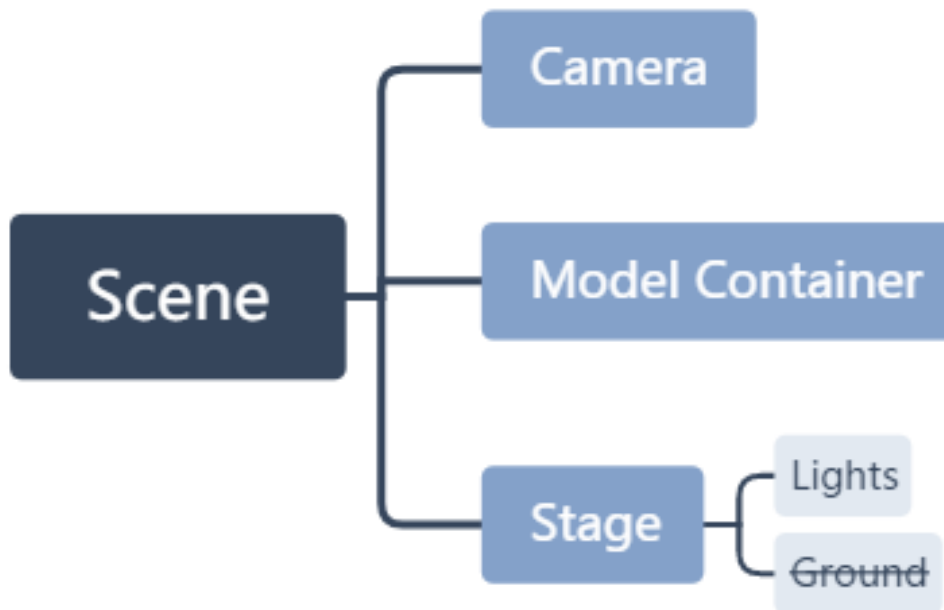


Ilustración 29 Árbol jerárquico de la escena 3D principal

Este diagrama, *Ilustración 29*, representa la disposición que tendrán los elementos en la escena 3D. Como se puede apreciar se trata de un diagrama simple, en el que se tienen 3 elementos “colgando” de la raíz. La cámara debe estar presente en un punto de la escena desde el que se renderizarán los demás elementos. El contenedor del modelo será el punto al que la cámara dirija su dirección.

El elemento “Stage” contendrá objetos adicionales como el conjunto de luces de la escena y un cubo aplanado que hará de “suelo”. Posteriormente se ha eliminado este último elemento al no aportar ninguna utilidad, consiguiendo dirigir con más fuerza la atención de los usuarios a lo realmente importante, los modelos 3D. La construcción de esta escena se encuentra descrita con más detalle en la primera iteración del proceso de implementación (*8.1 Iteración 1*).

7.3. Uso de patrones

Se usará el patrón de diseño “**Composite**” (componentes) ya que es el esquema seguido por el motor de videojuegos para gestionar los diferentes elementos de un proyecto.

Como anteriormente se ha mencionado, se usará el patrón **Modelo-Vista-Controlador** para implementar el control de la interfaz de usuario y el acceso a la información del modelo (*7.1.2 GUI Manager*). Cabe remarcar que este patrón no es de diseño, es arquitectónico, ya que expresa una organización estructural (responsabilidades e interrelaciones entre elementos o subsistemas) y tiene un mayor nivel de abstracción que un patrón de diseño.

8. Ejecución

Con una planificación lo suficientemente clara y con los recursos materiales disponibles (tablet y ordenador All-in-One) ya se puede empezar a implementar el sistema. En esta fase se detallarán todas las tareas realizadas, las dificultades encontradas en el proceso y las pruebas realizadas. Por cada funcionalidad se seguirá el tradicional esquema de actuación de tres pasos: diseño, implementación y pruebas. Las tareas van precedidas de su código de tarea, no deben ser confundidas con un índice literario.

8.1. Iteración 1

Grupos de Tareas a realizar: 0,1,2,3

Duración Estimada: 43

Fecha de Inicio Real: 06/03/2017

Fecha de Finalización Real: 29/03/2017

En esta iteración se procederá a la apertura del proyecto, creando el proyecto propio de Unity 3D. También se implementará la gestión de modelos 3D y texturas, se programarán las diferentes interacciones del usuario y se creará una estructura de proyecto para los modelos y su información.

0. Gestión de la Arquitectura de la Aplicación

0.1 Diseño

Una vez instalado Unity, para empezar a implementar el sistema serán necesarias 3 tareas. La primera será crear un proyecto en Unity 3D. Después de debe crear una escena principal sobre la que se añadirán los los elementos necesarios para el sistema. Posteriormente se guardarán los cambios, se cerrará Unity y se creará el proyecto GIT sobre la carpeta correspondiente.

0.2 Construcción del prototipo

Crear un proyecto en Unity resulta sencillo. Desde el panel principal de inicio de la herramienta se puede asignar un nombre y un directorio a un proyecto nuevo. Es **importante ajustar la configuración 3D** en lugar de 2D, ya que el motor realiza una configuración inicial del proyecto acorde a la selección. Esta configuración puede ser cambiada en cualquier momento de forma manual, cosa no recomendable ya que hay que modificar muchos elementos del proyecto como el motor de físicas, el motor de rendering, etc. No se añadirá ningún paquete de recursos adicionales.

El nombre elegido será **HCP DESKTOP** y el proyecto se creará en una partición secundaria del ordenador del desarrollador.



Ilustración 30 Nuevo proyecto en Unity 3D

La *Ilustración 30* muestra el aspecto del IDE al crear el nuevo proyecto. Como se puede apreciar en la imagen, se han creado manualmente una serie de carpetas para organizar eficientemente los diferentes elementos de trabajo que se usarán en el futuro (Fuentes, Imágenes, Scripts, Modelos, Materiales, Escenas y recursos en general).

0.3 Construcción de la escena

Al acceder a un proyecto recién creado, el IDE abre la plantilla de una escena, con una cámara y una luz direccional. En la pestaña “Hierarchy” puede apreciarse el árbol de la escena, con “Untitled” por nombre. Pulsaremos la opción File del menú principal y guardaremos esta escena en la carpeta “Scenes” anteriormente creada, con el nombre de “Main”.

Haciendo click derecho sobre el grafo de la escena añadiremos el objeto Canvas (representa la interfaz de usuario) que contendrá los menús y botones del sistema. También añadiremos un objeto vacío (llamado Origin) en las coordenadas origen del que colgará otro objeto vacío llamado ModelRoot que representará el componente Model Container (7.1.9 Model Container). Finalmente se creará otro objeto vacío llamado “Stage” que contendrá las luces. Se añadirán 4 luces direccionales con una rotación en el eje X de 45 grados y una diferencia de rotación en el eje Y de 90 grados cada una. Con esta colocación se conseguirá que el modelo quede iluminado desde arriba, desde todos los ángulos.



Ilustración 31 Escena principal construida

1. Gestión Modelos 3D y Texturas en tiempo real

Este grupo de tareas se ha dividido en: guardado de los modelos, el cargado de éstos y la gestión de las texturas.

1.1 Guardado de Modelos 3D

1.1.1 Diseño

Se deben guardar los modelos 3D y sus texturas de forma eficiente. Debido a las características propias de Unity este proceso no es trivial. El motor no está diseñado para cargar modelos y texturas desde disco directamente. Esta tarea sufrirá un importante retraso pues se han tenido que realizar pruebas de diferentes métodos de cargado. El problema es que Unity está pensado para, antes de usar recursos como texturas y objetos 3D, procesarlos y optimizarlos en la fase de compilación. Como estos modelos estarán

asociados a proyectos particulares del sistema, resulta inviable realizar una compilación del sistema con todos ellos.

Tras las pruebas realizadas ha propuesto una solución intermedia que aprovechará estas optimizaciones de unity sobre los modelos y evitará el procesamiento al cargarlos desde disco. El único punto negativo es que la creación de proyectos se deberá realizar desde el propio IDE de Unity. El método será el siguiente:

1. Se importarán las texturas y modelos como si de un asset de Unity se tratase. En este punto, el propio motor realiza sus optimizaciones correspondientes.
2. Se montarán estos elementos manualmente en la escena principal.
3. Se guardarán unas versiones serializadas en disco de los elementos.
4. Posteriormente, en el subgrupo de tareas siguiente a éste, se cargarán desde disco estos elementos y se crearán en memoria evitando un optimizado ya realizado.

1.1.2 Implementación

Unas de las optimizaciones que el motor realiza sobre un modelo 3D es el particionamiento del mismo en varios subconjuntos geométricos. Este hecho es importante ya que por cada modelo se guardará una colección de submallas.

Para realizar las funciones de serialización se ha creado un script llamado "Complex Mesh Serializer" y se ha añadido a la escena en un objeto vacío llamado "MeshParser". Este script tiene una referencia al componente ModelContainer para acceder a la geometría de un modelo cargado (y en un futuro cargar visualmente uno deserializado desde disco).

Adicionalmente a la serialización de las mallas, se han de guardar parámetros de posición, rotación y escalado del modelo para poder reconstruirlo correctamente en un futuro. Esta información se guardará en un archivo con extensión .cfg. La geometría del modelo (tabla hash de mallas) será serializada en un archivo .dat.

1.1.3 Pruebas de Unidad

Para probar la funcionalidad se ha dispuesto del modelo de una estatua Rakan creado por la empresa, *Ilustración 33*. Este modelo proporcionado es bastante ligero (unos 80 Megabytes y una textura de 10 Megabytes) lo que lo hace perfecto para realizar unas rápidas pruebas de serialización.

Tras las optimizaciones realizadas por Unity y la serialización, la información geométrica pasa a ocupar unos 12 Megabytes (**su tamaño se ha reducido un 85%**). La información extra (el archivo .cfg) ocupa 1 Kilobyte.

1.2 Cargado de Modelos 3D

1.2.1 Diseño

Se deben cargar los elementos anteriormente serializados y reconstruir el modelo correctamente.

1.2.2 Implementación

Para realizar esta area, se deben seguir los siguientes pasos, teniendo en cuenta el punto anterior.

1. Se deserializará el archivo .dat reconstruyendo la tabla hash que contiene las mallas. Se crearán estas mallas en la escena, usando un material por defecto “defaultMat” y sin texturas (por el momento).
2. Se deserializará la información general del proyecto presente en el archivo .cfg(posición, rotado del modelo, etc).
3. Se aplicarán las transformaciones pertinentes, reconstruyendo este modelo igual a como se encontraba en el momento de su serialización.

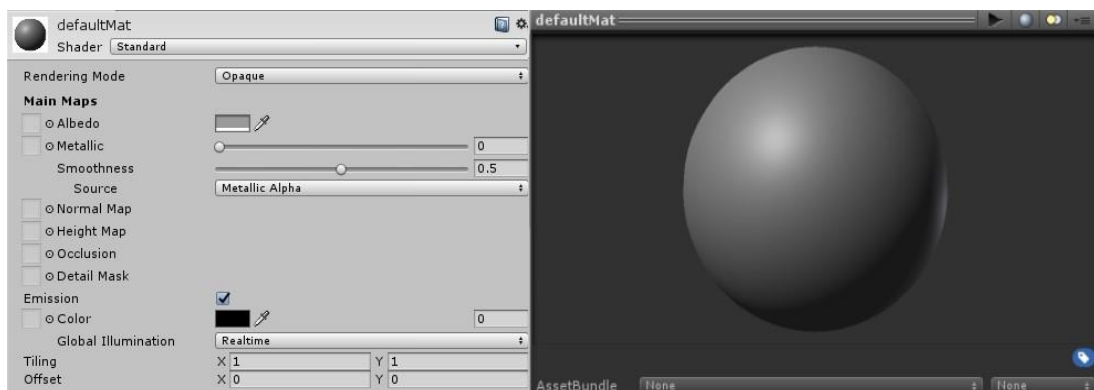


Ilustración 32 Material por defecto creado para los modelos

1.2.3 Pruebas de Unidad

Se ha usado el mismo modelo serializado del punto anterior para comprobar que la carga y aplicación de cambios se realiza correctamente, tanto a nivel de mallas como a nivel de la información general.



Ilustración 33 Cargado Rakan sin textura

1.3 Texturas y Materiales

1.3.1 Diseño

A partir del modelo creado en la escena se capturará la textura y se serializará la imagen que la representa. Posteriormente esta imagen se leerá desde disco como un paso adicional a la carga del modelo y sus transformaciones asociada.

1.3.2 Implementación

La textura del modelo (imagen) será guardada a disco directamente con el mismo nombre del modelo. No es necesario guardar un material del modelo ya que se usará uno por defecto llamado "defaultMat". El uso de materiales transparentes quedará pendiente para una futura mejora del sistema.

Se trabajará con texturas en formato JPEG por simplicidad. La textura se guarda en el mismo formato de imagen y ocupa exactamente lo mismo, ya que no se realizan compresiones sobre ella. Unity posee Métodos para leer imágenes y poder aplicar éstas como texturas.

1.3.3 Pruebas de Unidad



Ilustración 34 Cargado Estatua Rakan con textura

El modelo 3D de pruebas será el anteriormente usado rakan, *Ilustración 34*. Se han realizado varios procesos de guardado y carga para comprobar que efectivamente el modelo guarda, recupera y aplica las texturas correctamente.

1.4 Pruebas de Integridad

En esta tarea se ha comprobado que la gestión (guardado y carga) de modelos y texturas funciona correctamente. La implementación de las funcionalidades no ha afectado negativamente al comportamiento de las anteriores, consiguiéndose un proceso de completitud en la información guardada y recuperada.

2. Interacción Usuario

Este grupo de tareas tiene como objetivo crear métodos de interacción sobre los modelos a través de las diferentes entradas (ratón, teclado y pantalla táctil).

2.1 Entradas no Táctiles

Para las entradas no táctiles no necesitaremos ningún hardware específico más que el portátil en el que se está desarrollando el proyecto.

2.1.1 Ratón

2.1.1.1 Diseño

Las entradas a capturar en este punto y su finalidad serán:

- Scroll Rueda Ratón hacia arriba: Aumentar zoom al modelo.
- Scroll Rueda Ratón hacia abajo: alejar zoom del modelo.
- Arrastrar ratón en eje X positivo: Rotar modelo a la derecha.
- Arrastrar ratón en eje X negativo: Rotar modelo a la izquierda.
- Arrastrar ratón en eje Y positivo: Rotar modelo hacia arriba.
- Arrastrar ratón en eje Y negativo: Rotar modelo hacia abajo.

La interfaz deberá contener botones para realizar estas funciones dejándolos pulsados.

2.1.1.2 Implementación

Se han creado 2 scripts en este punto. El primero de ellos se llama "Extended Fly Cam" y su función será controlar los parámetros de la cámara tales posición y funciones de zoom. El segundo script se llama "DragOnTouch" y representará las entradas del ratón.

Este segundo script usa las funcionalidades expuestas por el primero, además de realizar las rotaciones pertinentes al modelo en sí. Se ha añadido un parámetro llamado “Sensibility” para ajustar la velocidad de las interacciones.

Se ha usado un pequeño plugin gratuito para realizar pequeñas animaciones dinámicas llamado [iTween](#). Gracias a este plugin podemos realizar animaciones basadas en interpolación de una forma muy fácil para dar un efecto más suave cuando se rota la figura para restablecer su posición inicial por ejemplo.

Se ha añadido un panel de botones, *Ilustración 35*, a la interfaz de usuario para realizar estas acciones con haciendo click sobre ellos. Estos botones disparan funciones en el “GUI Manager”, el cual ejecuta las funciones correspondientes.



Ilustración 35 Panel de botones de navegación de la GUI

2.1.1.3 Pruebas de Unidad

Se ha comprobado que efectivamente el modelo y la cámara responden a los eventos correspondientes y se ha ajustado el parámetro de sensibilidad a 80.

2.1.2 Teclado

2.1.2.1 Diseño

Las entradas a capturar en este punto y su finalidad serán:

- Tecla +: Aumentar zoom al modelo.
- Tecla -: alejar zoom del modelo.
- Tecla Dirección Derecha: Rotar modelo a la derecha.
- Tecla Dirección Izquierda: Rotar modelo a la izquierda.
- Tecla Dirección Arriba: Rotar modelo hacia arriba.
- Tecla Dirección Abajo: Rotar modelo hacia abajo.

2.1.2.2 Implementación

Estas entradas se han capturado en el script anterior llamado "DragOnTouch". No ha parecido necesario crear un simple script para esta pequeña parte.

2.1.2.3 Pruebas de Unidad

Todas las entradas de teclado funcionan y pueden usarse paralelamente a las entradas de ratón.

2.2 Entrada Táctil

Para llevar a cabo este grupo de tareas se debe disponer del ordenador all-in-one prestado por la universidad de Jaén.

2.2.1 Diseño

Las entradas a capturar en este punto y su finalidad serán:

- Arrastrar dedo en eje X positivo: Rotar modelo a la derecha.
- Arrastrar dedo en eje X negativo: Rotar modelo a la izquierda.
- Arrastrar dedo en eje Y positivo: Rotar modelo hacia arriba.
- Arrastrar dedo en eje Y negativo: Rotar modelo hacia abajo.

2.2.2 Implementación

Se puede pensar que estas interacciones pueden estar ya mapeadas por las mismas que el ratón, pero no es así. Los eventos “drag” sobre pantallas táctiles son diferentes a los eventos “drag” del ratón. Mientras que estos últimos trabajan con la entrada **Input.GetAxis** los eventos sobre pantallas táctiles usan **Input.GetTouch**. Se han añadido al script anterior ya que comparten funcionalidades con las demás entradas.

2.2.3 Pruebas de Unidad

Los 4 eventos son capturados correctamente y producen los efectos correctos, usando el parámetro de sensibilidad.

2.3 Pruebas de Integridad

Tras unas pruebas exhaustivas consistentes en usar todas las entradas por separado e ir mezclándolas podemos afirmar que los tres mecanismos de entrada funcionan correctamente y no provocan un funcionamiento anómalo entre ellos ni con las anteriores partes del sistema.

3. Proyectos

Este grupo de tareas tiene como objetivo la creación y cargado de los proyectos existentes en el sistema.

3.1 Creación

3.1.1 Diseño

Un proyecto debe englobar, un modelo 3D principal, sus texturas, los puntos de interés con la información multimedia y modelos 3D secundarios con sus texturas correspondientes.

La estructura básica de un proyecto será una carpeta que contenga toda la información:

- **Archivo .dat** del modelo principal, la geometría.
- **Archivo .cfg** con la configuración del modelo principal.
- **Subcarpeta textures**, con las texturas del modelo principal.
- **Subcarpeta zones**, necesaria para guardar los puntos de interés (se tratará más a fondo en [8.3 Iteración 2](#)).
- **Subcarpeta submeshes**, con los modelos secundarios (y sus texturas).

Los proyectos Unity utilizan un directorio en concreto, llamado “Persistent Data Path”. En sistemas windows este directorio se encuentra en “C:\Users\<USUARIO>\AppData\LocalLow\<EMPRESA>\<NOMBRE DEL PROYECTO>” (Díaz, 2017). Donde USUARIO es el nombre del usuario de Windows, EMPRESA es el nombre de la empresa (parámetro personalizable en el proyecto Unity, por defecto es DefaultCompany), y NOMBRE DEL PROYECTO es, en este caso concreto HCP DESKTOP. En sistemas Android se usa la carpeta “files” dentro del propio directorio de la aplicación.

Los modelos secundarios ofrecerán información como vistas más detalladas o texturas bajo infrarrojos o rayos X, por lo que no tendrán puntos de interés asociados.

Dado que se usará serialización de objetos y Unity la realiza de igual forma en windows y Android se asume que los proyectos serán portables.

3.1.2 Implementación

Se ha creado un método en el script “Complex Mesh Serializer” para generar toda la estructura cuando se cree un proyecto nuevo.

El proyecto se creará bajo un nombre dado. En caso de no proporcionar ningún nombre se usará el del modelo 3D. Esta creación de proyectos se usará ejecutando el sistema en el IDE de Unity 3D.

El script “Complex Mesh Serializer” proporciona unos atributos públicos accesibles para aportar información extra, como el nombre y diferentes texturas. Para conocer el uso de esta funcionalidad, consulte el anexo II ([15 Anexo II: Manual de creación de proyectos](#)).

Se han adaptado las anteriores funcionalidades de guardado de modelos y texturas para implementar esta tarea.

3.1.3 Pruebas de Unidad

Tras crear seis proyectos podemos comprobar que la estructura y los elementos necesarios se crean correctamente en la carpeta correcta.

3.2 Cargado

3.2.1 Diseño

El primer paso necesario es listar el conjunto de proyectos existentes en el sistema.

Una vez mostrados en un menú, el usuario elegirá el proyecto a cargar. Se deben realizar los mismos pasos que en la carga simple ([Gestión Modelos 3D y Texturas en tiempo real](#)).

3.2.2 Implementación

El sistema examina toda la carpeta del proyecto (PersistentDataPath) en busca de carpetas de proyecto. Una vez obtenida una lista con los nombres de las carpetas (nombre de los proyectos) se añadirán botones a un menú en la interfaz de usuario. Este menú, *Ilustración 36*, puede ser abierto y cerrado por medio de un botón “Proyectos” en la interfaz que siempre estará visible.



Ilustración 36 Menú proyectos

Cuando se selecciona un proyecto se carga su modelo principal y su textura. Si intentamos cargar un proyecto actualmente cargado el sistema no hace nada. Si ya existe un proyecto diferente cargado, el sistema reinicializa su estado, limpiando la geometría y liberando memoria por medio de la eliminación de texturas y la destrucción de objetos innecesarios.

La forma de incluir un modelo como modelo secundario de otro es tan fácil como copiar los archivos internos del primero (.dat, .cfg y carpeta textures) en la carpeta “submeshes” del segundo. Se ha programado un botón que muestra una lista de los modelos secundarios existentes en un proyecto,

pudiendo cargar éstos en la escena sustituyendo al principal. Este mecanismo es similar al del panel que muestra los proyectos existentes anteriormente descrito. El modelo principal siempre está cargado en escena, dado que éste suele ser el más usado y complejo. Los modelos secundarios son cargados cuando se requieren y eliminados de la memoria cuando se sustituyen por otro, optimizando los recursos.

3.2.3 Pruebas de Unidad

Se han creado y cargado tres proyectos diferentes, comprobando que el sistema funciona correctamente y que no se dejan en ejecución elementos de un proyecto anterior cuando se carga uno nuevo.

Se han incluido dos modelos secundarios en un proyecto y se ha comprobado que el cambio de modelo se ejecuta correctamente.

3.3 Pruebas de Integridad

Creando 2 proyectos desde cero y cargándolos varias veces se ha probado el correcto funcionamiento del sistema. Se han tenido que modificar las funcionalidades de la cámara, ya que al no tener un proyecto cargado, los eventos de entrada de usuario producían modificaciones que a la hora de cargar proyectos hacían que los modelos no se posicionasen correctamente.

Ahora, los eventos de rotación y zoom estarán bloqueados si no hay un proyecto abierto y al cargar un proyecto se reiniciarán los parámetros de la cámara. También se bloquearán los eventos si el menú de proyectos está abierto. Este mecanismo deberá extenderse a todos los posibles menús de la aplicación.

8.2. Evaluación iteración 1

Se ha compilado una versión de prueba del sistema (en windows) y se ha distribuido a los miembros del equipo junto con un par de proyectos para que prueben las interacciones y las cargas de los modelos y texturas. También se ha creado una temprana versión del manual de creación de proyectos, que finalmente se modificará hasta la versión final ([15 Anexo II: Manual de creación de proyectos](#)).

El “feedback” devuelto por los miembros es muy positivo, mostrando un sistema estable y funcional. La entrada táctil no ha podido ser probada por todos, ya que algunos miembros se encuentran fuera de las oficinas y no disponen de un dispositivo con pantalla táctil.

8.3. Iteración 2

Grupos de Tareas a realizar: 4

Duración Estimada: 38

Fecha de Inicio Real: 30/03/2017

Fecha de Finalización Real: 20/04/2017

Esta iteración tiene como objetivo desarrollar las tareas relacionadas con los puntos de interés y su información multimedia asociada.

4. Puntos de Interés

4.1 Gestión Puntos de interés

En este grupo de tareas se diseñará el sistema de puntos de interés y su funcionalidad.

4.1.1 Diseño

Este sistema deberá:

- Crear los puntos de interés
- Modificarlos
- Consultarlos
- Borrarlos

Un punto de interés se define como un área de puntos sobre la geometría de un modelo 3D que contiene información asociada. Esta información puede ser texto y/o multimedia. Para simplificar la implementación se trabajará con texto, y en el punto siguiente se tratará la información multimedia.

Un punto de interés tendrá un identificador (numérico), un nombre, una descripción y una lista de puntos que definan el área. Estas áreas deben representarse visualmente mientras se están definiendo y para consultar la información asociada. Se requiere de una lista general de puntos de interés en la interfaz para poder acceder rápidamente a uno en concreto.

Como se ha mencionado en las tareas relativas a la gestión de proyectos de la iteración 1, los modelos secundarios no tendrán puntos de interés asociados (aunque de tenerlos, gracias a la modularidad de los archivos en disco sería fácil extender esta parte en un futuro).

4.1.2 Implementación

Se ha creado un script llamado “Zone Manager” para implementar todas las tareas de esta parte. Dada su estrecha relación con los proyectos se ha asignado al objeto de la escena MeshParser. Este script gestionará todo lo referente a puntos de interés. En él se define el modelo del punto de interés (Zone) y realiza las operaciones definidas en la etapa de diseño.

Para crear y visualizar estos puntos de interés se necesitan elementos gráficos simples. Con ellos se compondrá su geometría, evitando guardar una malla completa.

Elementos Geométricos

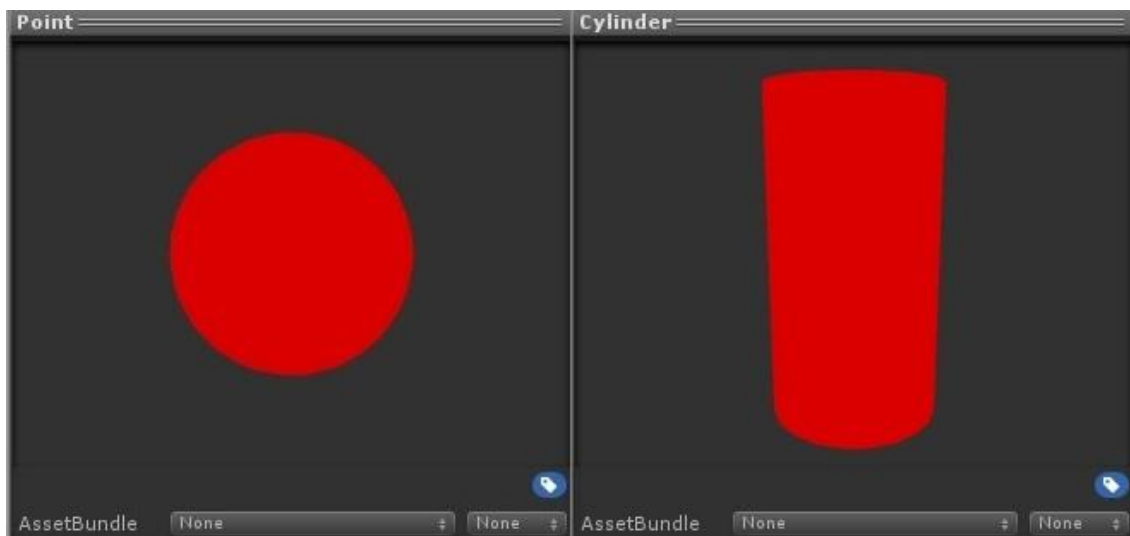


Ilustración 37 Elemento punto y cylinder de los puntos de interés

Con estos elementos, *Ilustración 37*, se creará una forma geométrica a partir de una colección de puntos 3D. Cada punto será representado por el elemento punto y 2 puntos consecutivos serán unidos por el elemento cilindro. Estos elementos pueden parecer 2D, pero son 3D. Esto es debido al shader que está usando el material, que le da un aspecto más uniforme, sin brillos o sombras.

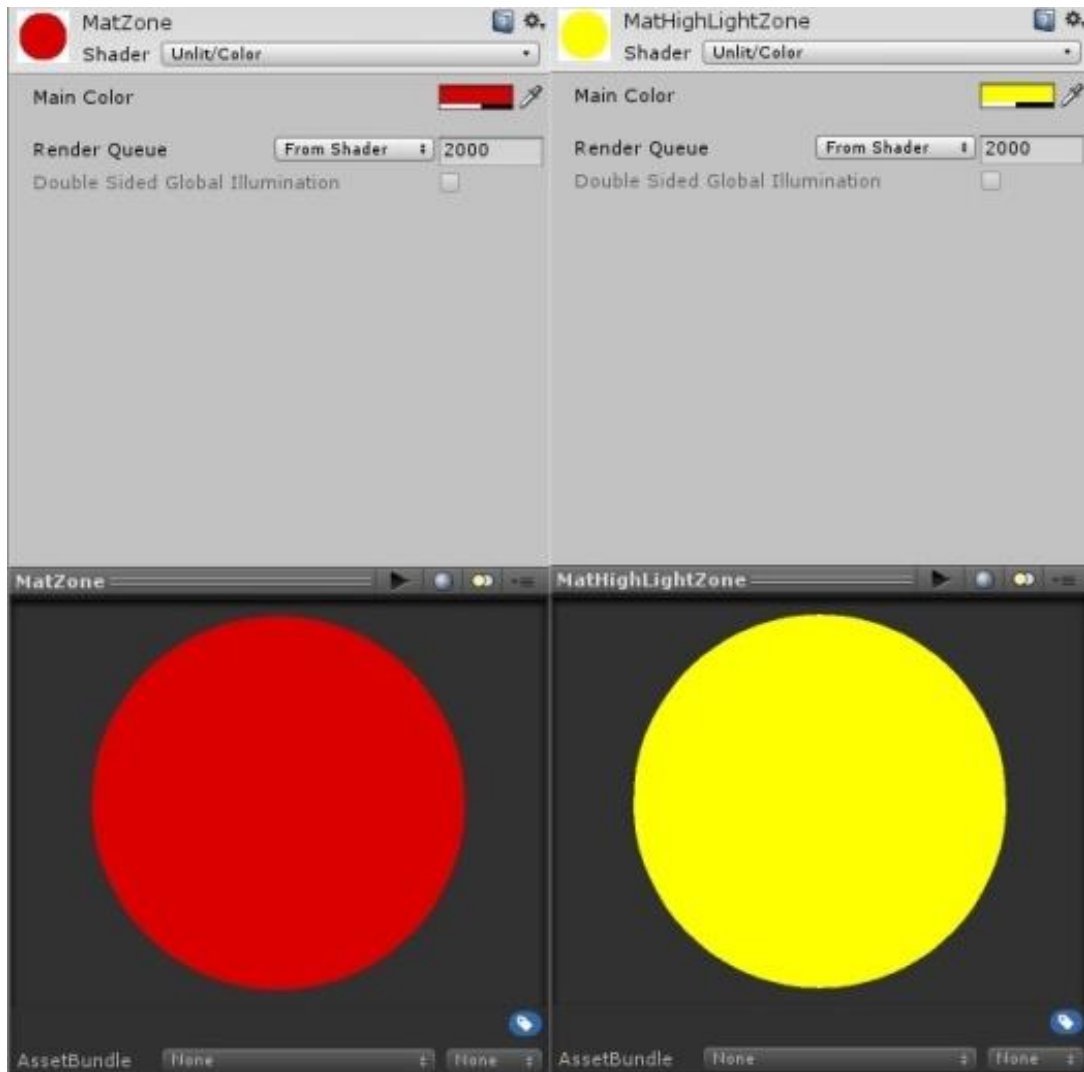


Ilustración 38 Materiales de la geometría de los puntos de interés

Adicionalmente se ha creado un material que resalte más el área, para cuando se seleccione un punto de interés.

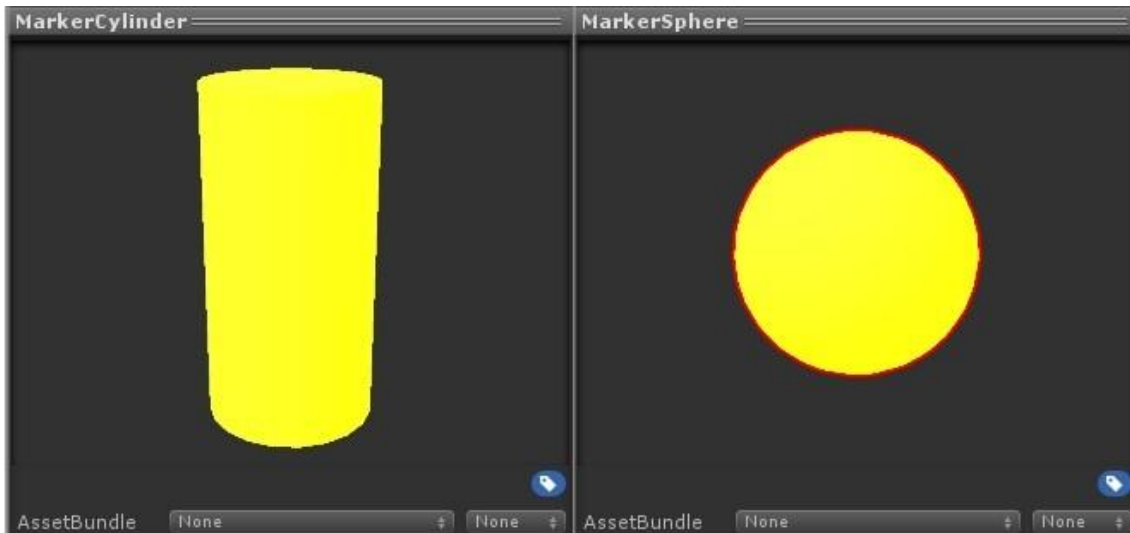


Ilustración 39 Elemento cylinder y sphere del marcador de puntos de interés

Para visualizar las zonas y acceder a su información con más facilidad se colocarán unos marcadores 3D creados con esta geometría, *Ilustración 39*. Los materiales usados pueden verse en la siguiente *Ilustración 40*.

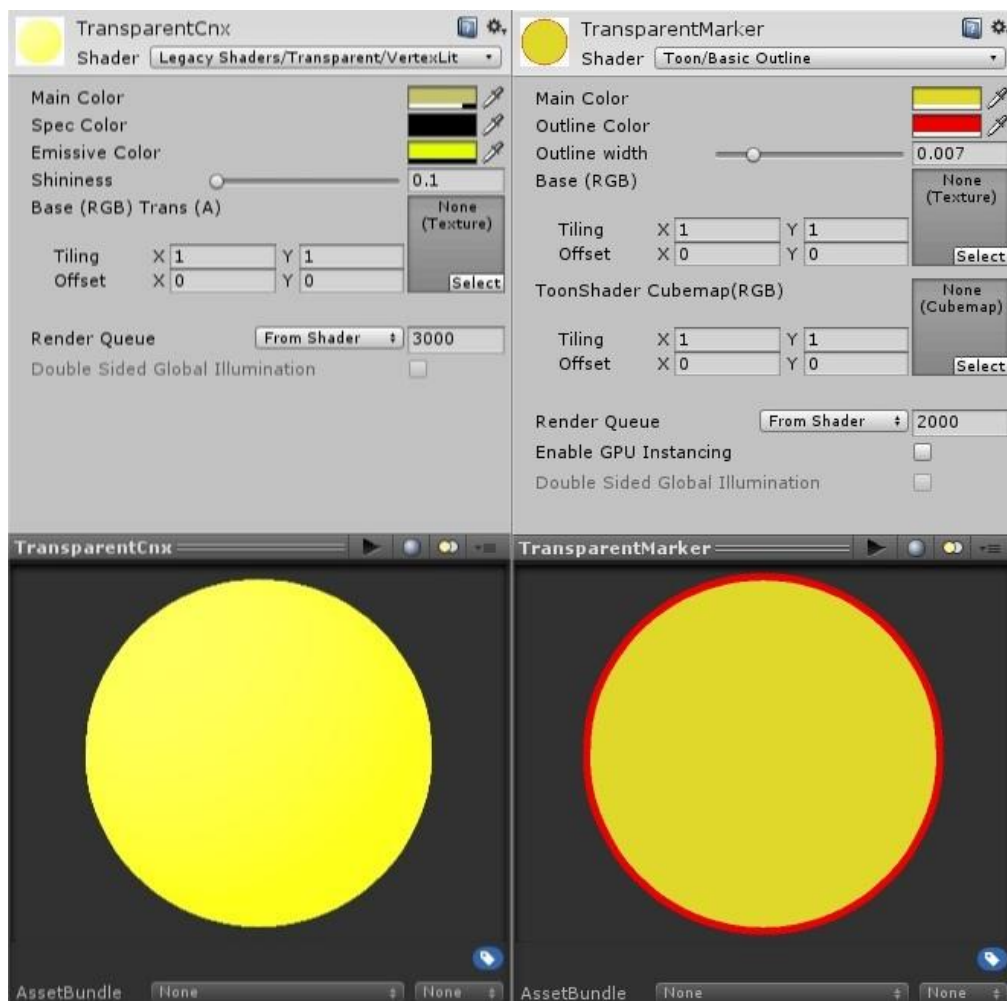


Ilustración 40 Materiales del marcador de puntos de interés

Creando los puntos

A través de un botón (“Crear”) en la interfaz se habilita la creación de puntos de interés sobre el modelo. Se recomienda usar la función bloquear (freeze) de la cámara para que, al arrastrar el ratón sobre la figura ésta no se mueva. Como se puede apreciar en la siguiente imagen, se van creando pequeños puntos rojos sobre el modelo. Al soltar el ratón podremos completar este punto de interés con información o descartarlo a través de un panel que se presenta en el siguiente apartado. Este efecto de “dibujar” sobre el modelo se obtiene usando un “collider” adaptado al modelo que genera eventos y proyectando la posición del ratón desde la pantalla 2D al modelo.



Ilustración 41 Creando la geometría de un punto de interés

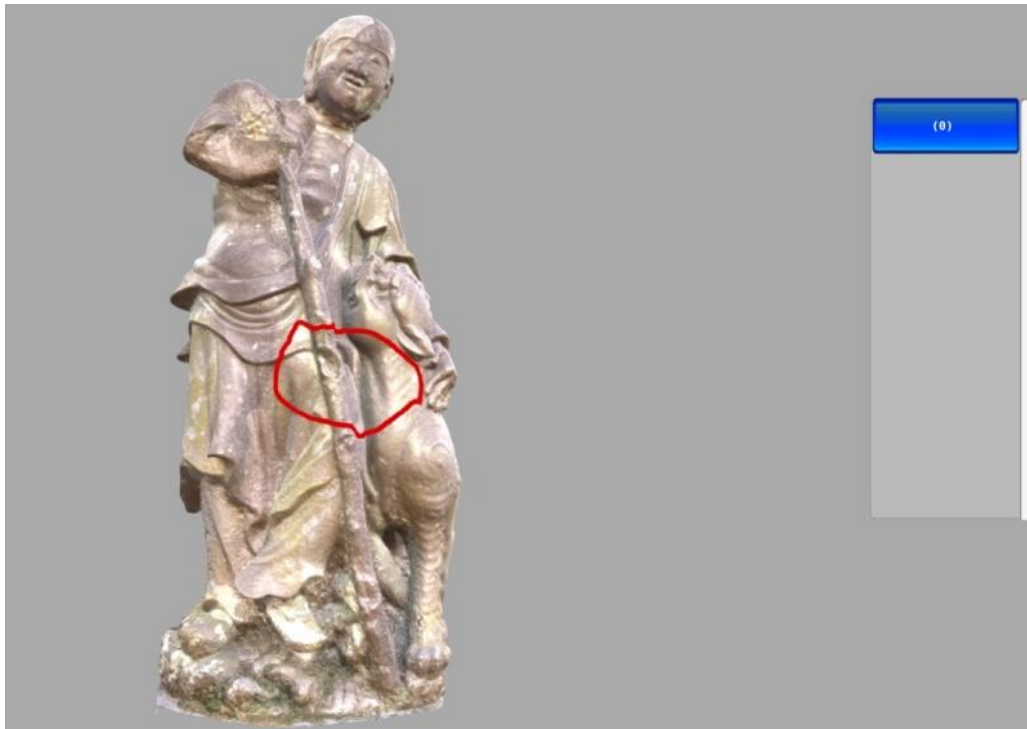


Ilustración 42 Punto de interés creado

Cuando se termine de crear el punto de interés aparecerá en una lista lateral. Como el punto se ha creado sin nombre, aparecerá su id asignada por el sistema. Pulsando sobre la lista o la geometría accederemos al panel de información. También se agregará un marcador al punto para distinguirlo temporalmente de los demás.

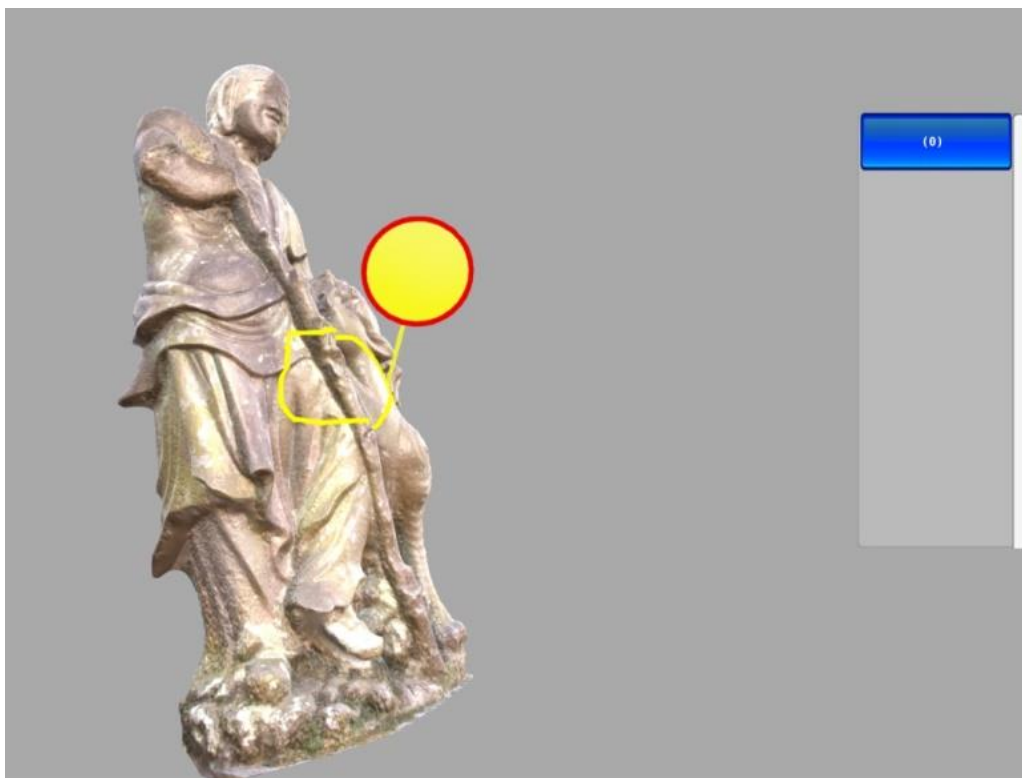


Ilustración 43 Punto de interés seleccionado

El Panel

Este panel se ha creado para completar la información a la hora de crear un punto de interés y para mostrarla a los usuarios que seleccionen el punto. Es un panel simple, que cuando muestre la información deshabilitará los botones para modificarla. En el script "Zone manager" existe un atributo para deshabilitar la creación y edición de zonas, así se puede compilar una versión sólo para visitantes y otra para administradores que necesiten editar la información.



Área de Interés

ID

Id

NOMBRE

Nombre

DESCRIPCIÓN

Descripción

Guardar Borrar Cerrar

Ilustración 44 Panel de Puntos de Interés básico

Persistencia

Como anteriormente se mencionó en las tareas de proyectos ([8.1 Iteración 1](#)), existe una carpeta “zones” donde se guardará la información relativa a los puntos de interés. Esta información será una tabla hash usada para almacenarlos y se serializará a un archivo .zones. Se ha agregado un botón a la interfaz para guardar el conjunto de puntos de interés y sus cambios (creaciones, modificaciones y borrados). Al iniciar un proyecto, si éste posee puntos de interés definidos, se cargarán y se habilitará la lista lateral.

4.1.3 Pruebas de Unidad

Se ha comprobado que los puntos se crean correctamente, que se guardan, se recuperan de disco y se eliminan. También se ha probado que la información asociada sólo puede cambiarse si está habilitada la opción de creación y edición.

Tras varios problemas a la hora de representar las áreas en su posición correspondiente se han realizado ajustes en la creación de la geometría de los puntos de interés para que éstas se guarden en su correcta posición relativa al modelo 3D.

4.2 Multimedia

4.2.1 Diseño

Se entiende por contenido multimedia el conjunto de imágenes, videos y audios que el sistema deberá manejar. Buscando formas de cargar estos datos eficientemente desde disco se vuelven a encontrar dificultades. Los videos y audios son procesados por Unity desde su IDE, por lo que cargarlos en tiempo de ejecución desde disco no es una tarea ni fácil ni rápida. En cambio, las imágenes tienen un procesamiento muy ligero, por lo que se pueden cargar en tiempo real de forma fácil.

Como se tiene un tiempo bastante ajustado, estas tareas van a centrarse en la gestión de imágenes por punto de interés. Dejando para una futura actualización el tratamiento de videos y audios. Esta decisión ha sido tomada por el equipo de la empresa al completo.

Se deberá implementar una forma cómoda de añadir imágenes a un punto de interés y la funcionalidad necesaria para mostrarlas cuando se consulte el punto.

4.2.2 Implementación

Para realizar la inclusión de las imágenes en los puntos y la posterior carga de éstas desde disco se creará un script llamado "Image Loader". Este script estará asociado al objeto Mesh Parser, dada su estrecha relación con el script "Zone Manager". Para elegir los archivos desde el disco, se usará un plugin gratuito de Unity llamado [FileBrowser](#). Cabe destacar que este plugin es multiplataforma pudiendo usarse en compilaciones para Windows y Mac OS.

Al cargar un proyecto, se configura el script para trabajar sobre su directorio . Cuando se selecciona un punto se cargan sus imágenes asociadas. Estas imágenes se encuentran en la carpeta "zones" del proyecto, organizadas por la id del punto de interés al que están asociadas.

Se puede añadir una nueva imagen (en .png, .jpg o .jpeg) marcándola en la ventana de selección de archivos (gracias al plugin). Para visualizar las imágenes se ha construido un panel, [Ilustración 45](#), con unos botones laterales que servirán para navegar por las diferentes imágenes a modo de carrusel. Este panel también mostrará la información textual añadida en la creación del punto.

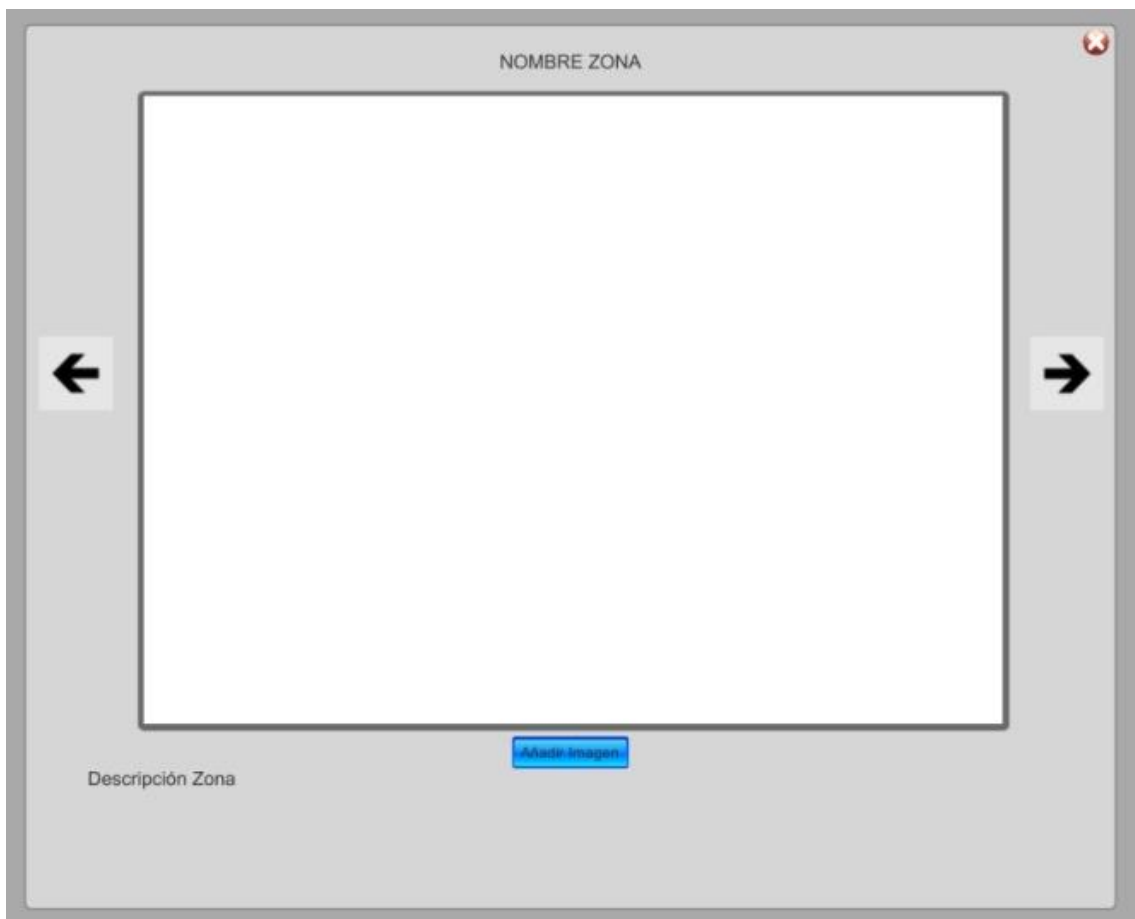


Ilustración 45 Nuevo panel de visualización de Punto de Interés

4.2.3 Pruebas de Unidad

Se ha comprobado que las imágenes se copian correctamente a la carpeta del proyecto, que se cargan y visualizan correctamente al consultar un punto de interés. También se ha comprobado que al borrar un punto de interés se elimina su carpeta de recursos asociada (imágenes, en un futuro contendrá más tipos de archivos multimedia).

4.3 Pruebas de Integridad

Las funcionalidades desarrolladas en esta iteración (gestión de puntos de interés y multimedia) están muy interrelacionadas, por lo que al realizar las pruebas de unidad de las últimas se ha comprobado el funcionamiento conjunto con las primeras.

Se ha creado un proyecto desde cero con el modelo de pruebas “rakan”, siguiendo el manual de creación de proyectos ([15 Anexo II: Manual de creación de proyectos](#)). Después se han definido unas cuantas zonas de interés con imágenes asociadas. Todo funciona como se esperaba. A continuación se muestran unas imágenes del proceso.



Ilustración 46 Creando Puntos de Interés 1

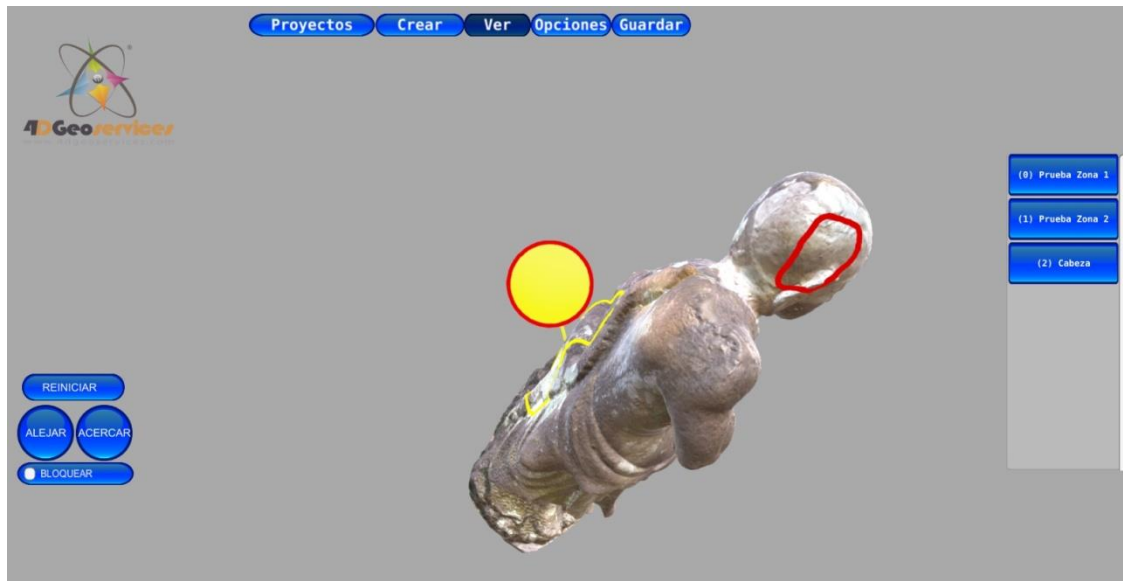


Ilustración 47 Creando Puntos de Interés 2

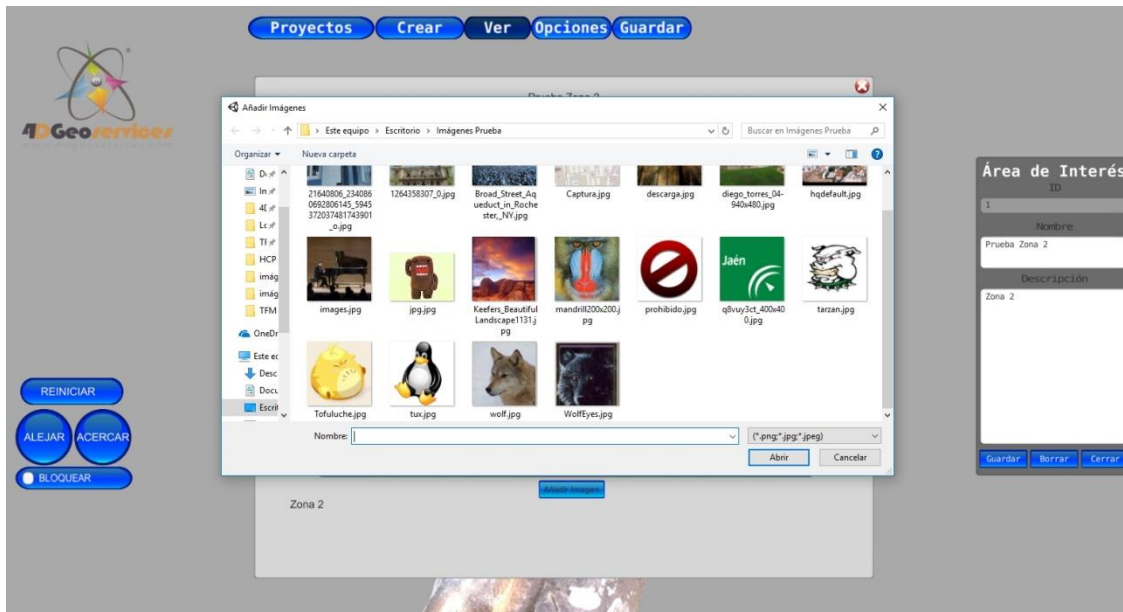


Ilustración 48 Añadiendo una imagen a un Punto de Interés

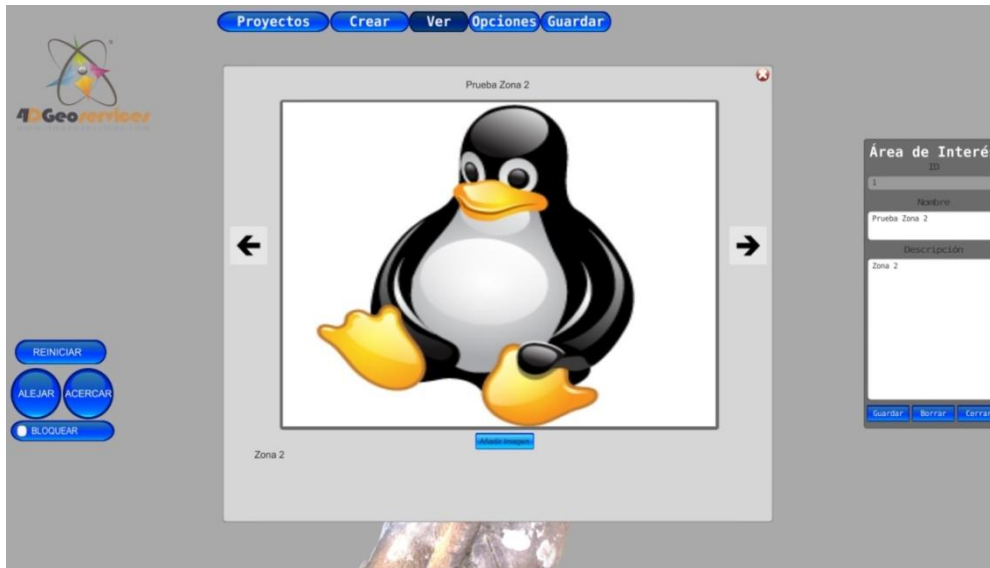


Ilustración 49 Mostrando la imagen anteriormente añadida

8.4. Evaluación iteración 2

Se ha compilado una versión de prueba del sistema (en windows) y se ha distribuido a los miembros del equipo junto con un proyecto de prueba del modelo “rakan”. El objetivo será probar la gestión de puntos de interés y sus imágenes asociadas. Se ha modificado y completado el manual de creación de proyectos ([15 Anexo II: Manual de creación de proyectos](#)).

No se ha reportado ningún problema reseñable con esta versión del sistema. Algunos apuntes realizados por el equipo son:

- Los marcadores se crean automáticamente dependiendo de la geometría del punto de interés. Definir una geometría que envuelva al modelo por varias caras creará un marcador en sitios extraños o que atraviese la figura, cosa no deseable. *Se ha cambiado la forma del generar el marcador teniendo en cuenta un eje central del modelo para que esto no ocurra.*
- Debido a cómo se realiza la gestión de imágenes (por carpeta) podemos cargar múltiples imágenes en un punto de interés copiándolas a la carpeta de ese punto. Este hecho puede resultar útil para incluir grandes lotes de imágenes de una sola vez.
- Al existir desacople entre partes de un proyecto en disco, se pueden hacer copias de seguridad, por ejemplo, sólo de las zonas. Este hecho puede tener un gran potencial en la parte de conservación de patrimonio ([1.4 Motivación](#)) y se estudiará su uso.

8.5. Iteración 3

Grupos de Tareas a realizar: 5,6,7,8

Duración Estimada: 38

Fecha de Inicio Real: 21/04/2017

Fecha de Finalización Real: 23/05/2017

Al finalizar esta iteración se tendrán implementados los siguientes grupos de tareas:

- **Plataformas Objetivo**, probando y adaptando todas las funcionalidades anteriores a Android si fuese necesario.
- **Internacionalización**, brindando al sistema un mecanismo para traducción y carga de idiomas.
- **Customización de la Escena**, personalizando la escena principal a gusto del usuario
- **Cierre del proyecto**, generando los ejecutables requeridos.

5. Plataformas Objetivo

5.1 Diseño

El objetivo de estas tareas consiste en obtener una versión del sistema capaz de funcionar sobre Android. Tras un análisis previo al proyecto sobre las peculiaridades de Unity 3D, no resulta tan trivial como se podría pensar generar un sistema homólogo en Android. El procedimiento a seguir será el siguiente:

1. Realizar la generación de la apk.
2. Comprobar toda la funcionalidad. Si algo no funciona se arreglará y adaptará.
3. Repetir el proceso hasta obtener una versión totalmente funcional.

La versión mínima requerida del sistema operativo será Android 4.4.4 y la máxima Android 7.0.

5.2 Adaptación

Generación de la APK

Para generar la APK debemos cambiar la plataforma objetivo del proyecto Unity. Es requisito indispensable disponer del SDK de Android en el sistema, con las versiones correspondientes descargadas y una versión del Java JDK. Para que Unity sea capaz de compilar proyectos en Android se debe facilitar la ruta del SDK y JDK en “Edit” > “Preferences”, *Ilustración 50*.

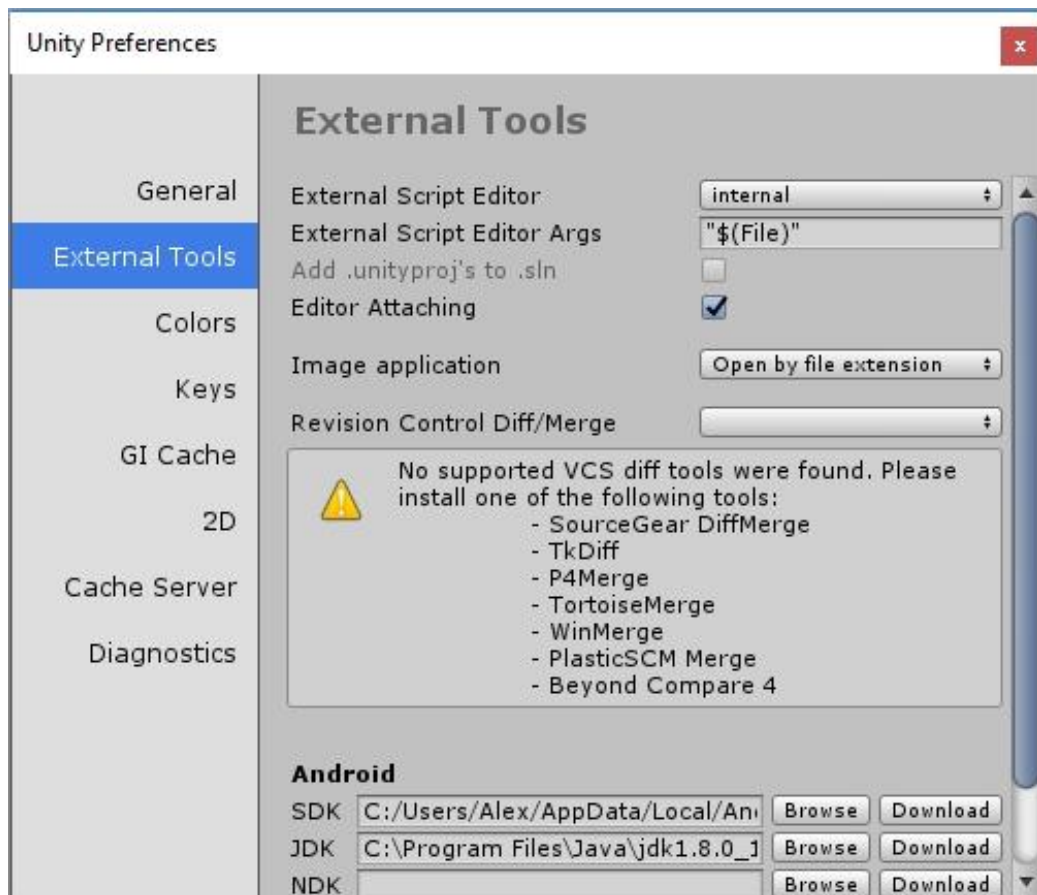


Ilustración 50 Unity Preferences

En el menú principal “File” > “Build Settings”, *Ilustración 51*, se encuentra la siguiente vista. Seleccionando la plataforma Android, añadiendo las escenas a incluir y marcando la escena Main como escena de comienzo ya se tendrá configurada la plataforma objetivo.

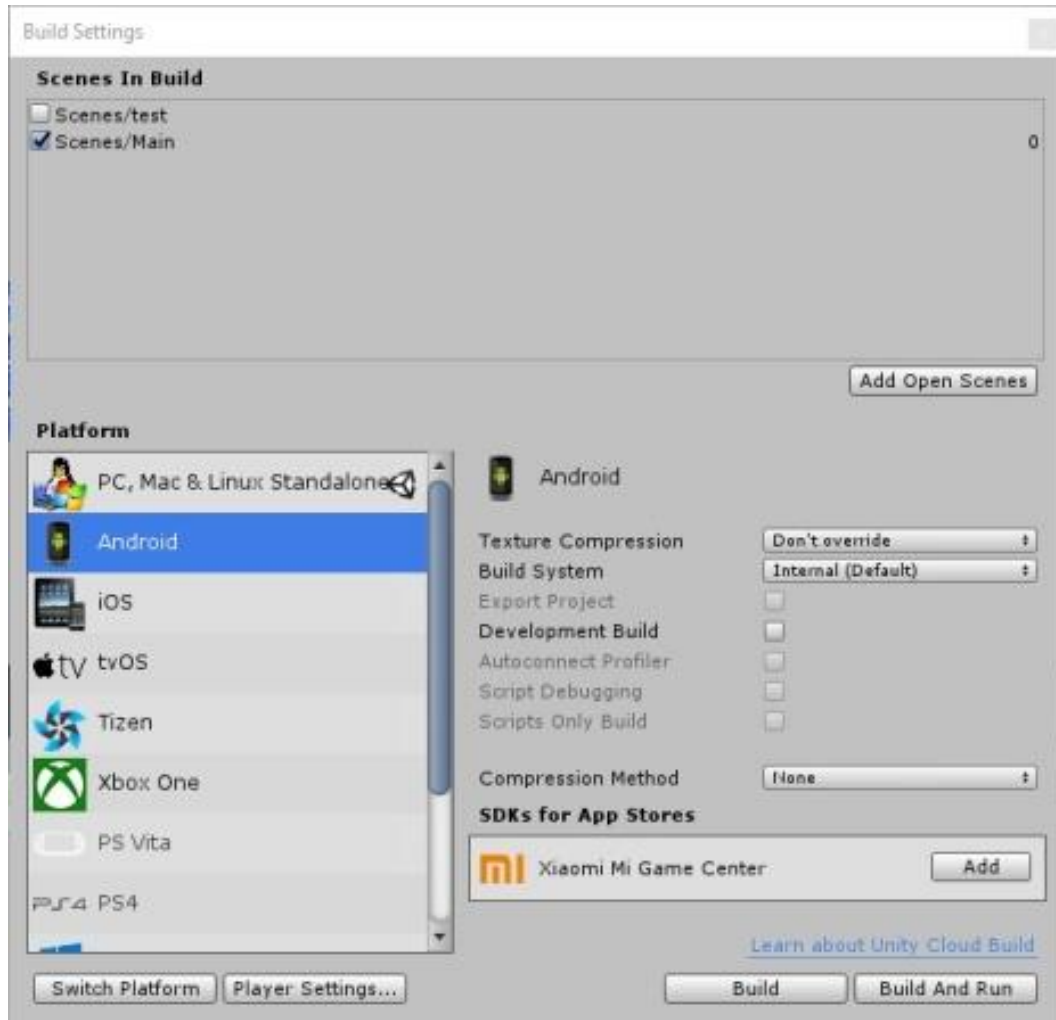


Ilustración 51 Cambio de Plataforma Objetivo

Adicionalmente, la primera vez que se realice este cambio se deberá aportar información extra pulsando sobre el botón “Player Settings” de la ventana anterior.



Ilustración 52 Configuración del proyecto Android

En esta ventana, *Ilustración 52*, podremos configurar la información necesaria para la correcta compilación del proyecto en Android. Una vez realizados todo estos pasos ya podremos construir la apk.

Ejecución inicial

Tras generar la apk se ha instalado en la tablet Samsung Galaxy Tab A. Se ha copiado un proyecto anteriormente generado y se ha iniciado la aplicación. Tal y cómo se dijo en las tareas de creación de proyectos, los proyectos son portables y no se tiene ningún problema al leer los archivos y deserializarlos.

Dado que en Android el hilo de la UI (principal) está separado de los demás procesos subyacentes, los botones y eventos sobre la pantalla siguen funcionando mientras se realizan largas tareas de carga de modelos 3D. Este hecho da como resultado funcionamientos anómalos. Un cambio será bloquear cualquier interacción de usuario cuando se tenga una carga de modelo.

Al abrir modelos secundarios la aplicación se cierra inesperadamente. Consultando los archivos de log y monitorizando el sistema se observa que la actividad del sistema intenta usar muchos recursos, por lo que es eliminada. Tras una depuración exhaustiva el problema es localizado en la carga de texturas. Una imagen de 12 Megabytes cargada como textura utilizando el proceso actual da como resultado una textura en memoria de 576 Megabytes. Siendo la imagen de 12.000 por 12.000 píxeles se tienen 144 millones de píxeles. Cada píxel ocupa 4 bytes (debido a los canales R G B y alfa) dando como resultado 576 Megabytes, lo que es más de la mitad de la memoria ram disponible en la tablet.

Las interacciones del usuario hacen que el modelo se mueva demasiado rápido. Hay que ajustar el parámetro de sensibilidad.

Cambio: Bloqueo de Eventos

Para arreglar el problema de las eventos de la interfaz se bloquearán todas las entradas de usuario cuando se está cargando un modelo, cuando esta carga termine se desbloquearán. Se ha incluido un panel de espera con un mensaje para ofrecer “feedback” al usuario.

Cambio: Optimización de Texturas

Se han investigado nuevas formas de cargar las texturas sin un buen resultado. Ahondando en este tema se ha llegado a una solución intermedia que evita usar tantos recursos en memoria. Unity posee algoritmos de compresión para texturas. Estos algoritmos se suelen usar en memoria para evitar que el sistema se colapse. Unity los usa automáticamente si es el motor quien gestiona las texturas. Como no es el caso, ya que la cargamos dinámicamente desde disco, se procederá de la siguiente forma.

A la hora de crear el proyecto, en lugar de guardar la textura, se creará una versión comprimida de ésta usando el algoritmo de Unity en formato DXT1 (o DXT5 si tiene canal alfa). Los bytes de esta textura comprimida se serializarán en un archivo (. ctx) junto con otros datos extra como el tamaño y el nombre del proyecto al que pertenece entre otros.

Cuando se necesite cargar la textura se deserializará la versión comprimida directamente y se aplicará al modelo.

Usando la anterior textura de 12 Megabytes preprocesada y comprimida se tiene que:

1. En disco ésta ocupa 91´5 Megabytes aproximadamente. Es decir, su tamaño es más de 7 veces superior.
2. En memoria ocupa lo mismo que en disco, por lo que pasa de necesitar 576 a 91,5 Megabytes, **reduciendo su cantidad requerida de memoria RAM en un 84%**.

Cambio: Sensibilidad de Rotación y Zoom

Se ha ajustado el parámetro de sensibilidad hasta conseguir un movimiento del modelo3D más lento y controlable.

5.3 Pruebas

Se ha comprobado que los cambios realizados en la tarea anterior hacen que el sistema funcione en su totalidad en Android. Se ha creado un proyecto limpio con el modelo de pruebas "rakan" y se ha copiado en la tablet. A éste proyecto se le han añadido y borrado puntos de interés exitosamente. Las imágenes asociadas a puntos de interés previos se han visualizado correctamente.

6. Internacionalización

Para internacionalizar el sistema se definirán los ficheros de idioma y cómo el sistema los aplicará.

6.1 Ficheros de idioma

6.1.1 Diseño

Ya que la facilidad en la creación e inclusión de nuevos idiomas al sistema es un factor muy importante se usarán formatos ligeros de intercambio de información ampliamente conocidos. En este caso los ficheros de idioma seguirán el formato JSON.

6.1.2 Creación Ficheros

Como Unity permite asignar un nombre a cada elemento del sistema, usaremos la estructura clave valor típica del formato JSON para almacenar en el archivo las cadenas con el formato elemento cadena, en el idioma correspondiente. Los nombres de ficheros seguirán la convención de códigos propuesta por la ISO 639-1 (Korpela, 2006). Inicialmente se han creado 2 ficheros: **"en_lang.json"** será el nombre del fichero inglés y **"es_lang.json"** será el del idioma español.

```
{
  "projects": "Projects",
  "create": "Create",
  "view": "View",
  "save": "Save",
  "exit": "Exit",
  "about": "About",
  "zoomIn": "ZOOM +",
  "zoomOut": "ZOOM -",
  "restart": "RESTART",
  "freeze": "BLOCK",
  "delete": "Erase",
  "close": "Close",
  "zoneTitle": "Zone of Interest",
  "name": "Name",
  "name_placeholder": "Name",
  "description": "Description",
  "description_placeholder": "Description",
  "langTitle": "Languages",
  "submeshtitle": "Models",
  "submeshbutton": "Models"
}
```

Ilustración 53 Fichero JSON "en_lang.json"

6.2 Módulo Internacionalización

6.2.1 Diseño

Se creará un script que lea los ficheros JSON de idiomas desde disco y aplique los cambios a los elementos de la interfaz de usuario.

Los archivos creados en el punto anterior deberán encontrarse en una carpeta llamada “lang” creada en el directorio “Persistent Data Path” de la aplicación.

En sistemas windows este directorio se encuentra en “C:\Users\<USUARIO>\AppData\LocalLow\<EMPRESA>\<NOMBRE DEL PROYECTO>” (Díaz, 2017). Donde USUARIO es el nombre del usuario de Windows, EMPRESA es el nombre de la empresa (parámetro personalizable en el proyecto Unity, por defecto es DefaultCompany), y NOMBRE DEL PROYECTO es, en este caso concreto HCP DESKTOP. En sistemas Android se usa la carpeta “files” dentro del propio directorio de la aplicación.

Por defecto y si ocurre algún fallo en este módulo del sistema, el idioma que se cargará será el inglés.

6.2.2 Implementación

Se ha creado un script llamado “Internalization Manager” que contiene referencias a todos los elementos traducibles de la UI. Es script ha sido agregado al objeto “Canvas”, el cual representa la interfaz de usuario. Este script traducirá directamente la aplicación detectando el idioma del sistema operativo en el que se encuentre al iniciar la ejecución. Examinará el directorio “lang” para obtener una lista de los idiomas disponibles.

Unity ofrece funcionalidades para crear, leer y manipular ficheros JSON de manera muy cómoda, por lo que la lectura de un fichero de idioma es trivial. El único detalle a tener en cuenta es que se debe usar una codificación UTF-8 para que la conversión de bytes a cadenas de texto se realice correctamente.

Se han agregado 2 paneles a la interfaz y un botón de opciones. A través de ellos se permitirá al usuario cambiar de idiomas cómodamente.



Ilustración 54 De izquierda a derecha, Panel de Opciones y Panel de Selección de Idiomas

A partir de este punto, todos los nuevos elementos que se creen en la interfaz que sean traducibles se agregarán al script y a los ficheros de idioma correspondientes.

6.2.3 Pruebas de Unidad

Se han realizado pruebas sobre los ficheros de idiomas, borrando algunos campos para comprobar la tolerancia del sistema frente a fallos y que la inexistencia de una cadena de traducción en concreto no pare todo el proceso, afectando únicamente al elemento en cuestión que no será traducido. La lista dinámica de idiomas disponibles en el panel se crea correctamente y permite al usuario cargar un idioma en concreto. Si se intenta volver a aplicar el idioma aplicado actualmente no se realizará ninguna acción.

7. Customización Escena

En este grupo de tareas se implementarán las modificaciones de la escena principal. Estas modificaciones son, el cambio de intensidad lumínica y el cambio del color de fondo de la escena.

7.1 Intensidad Luces

7.1.1 Diseño

La escena posee 4 luces direccionales, creadas en el grupo de tareas 0.3 Construcción de la escena. Se debe crear un mecanismo que permita controlar la intensidad de los 4 elementos simultáneamente.

7.1.2 Implementación

Se creará un script llamado "Light Manager" que controlará las 4 luces de la escena. Este script posee atributos para definir una intensidad máxima (1.5), una mínima (0) y un valor por defecto (porcentaje entre estos dos valores, 0.37, osea un 37%). Adicionalmente, en el panel de opciones se ha incluido un slider para poder cambiar la intensidad.

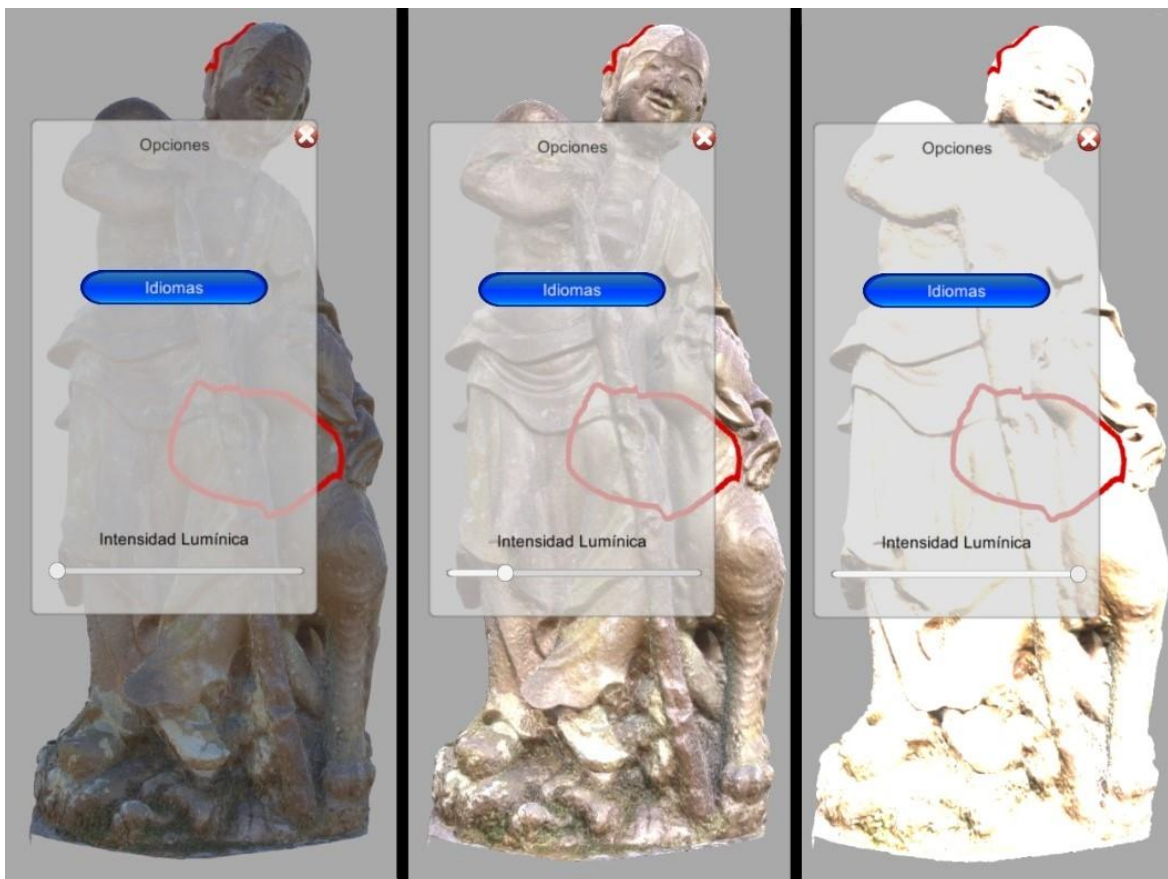


Ilustración 55 Diferencia de intensidades sobre el modelo "Rakan"

7.1.3 Pruebas de Unidad

Se han realizado pruebas y el slider funciona correctamente, cambiando los valores de intensidad a través del script "Light Manager". Esta configuración no se guarda al cerrar la aplicación y volver a abrirla, como se esperaba.

7.2 Color Fondo

7.2.1 Diseño

El primer paso es encontrar una forma en la que el usuario pueda especificar un color fácilmente. Aplicar un color dado al fondo será fácil a través de algún script que tenga una referencia al elemento skybox de la escena.

7.2.2 Implementación

Para permitir a los usuarios elegir un color se ha usado un plugin gratuito de Unity llamado [HSV Color Picker](#). Este plugin es un panel que permite definir colores desde valores hexadecimales, valores RGB o un cuadro interactivo HSV. Se ha creado un botón en el panel de opciones para mostrar este panel de selección de color.

Para cambiar el color del skybox de la escena se ha creado un pequeño script que contiene una referencia al skybox (Background Color Manager) asociado al script propio del plugin. Como el script del plugin lanza eventos cuando el color definido en el panel cambia, se llamará a un método del Background Color Manager que cambiará finalmente el color.

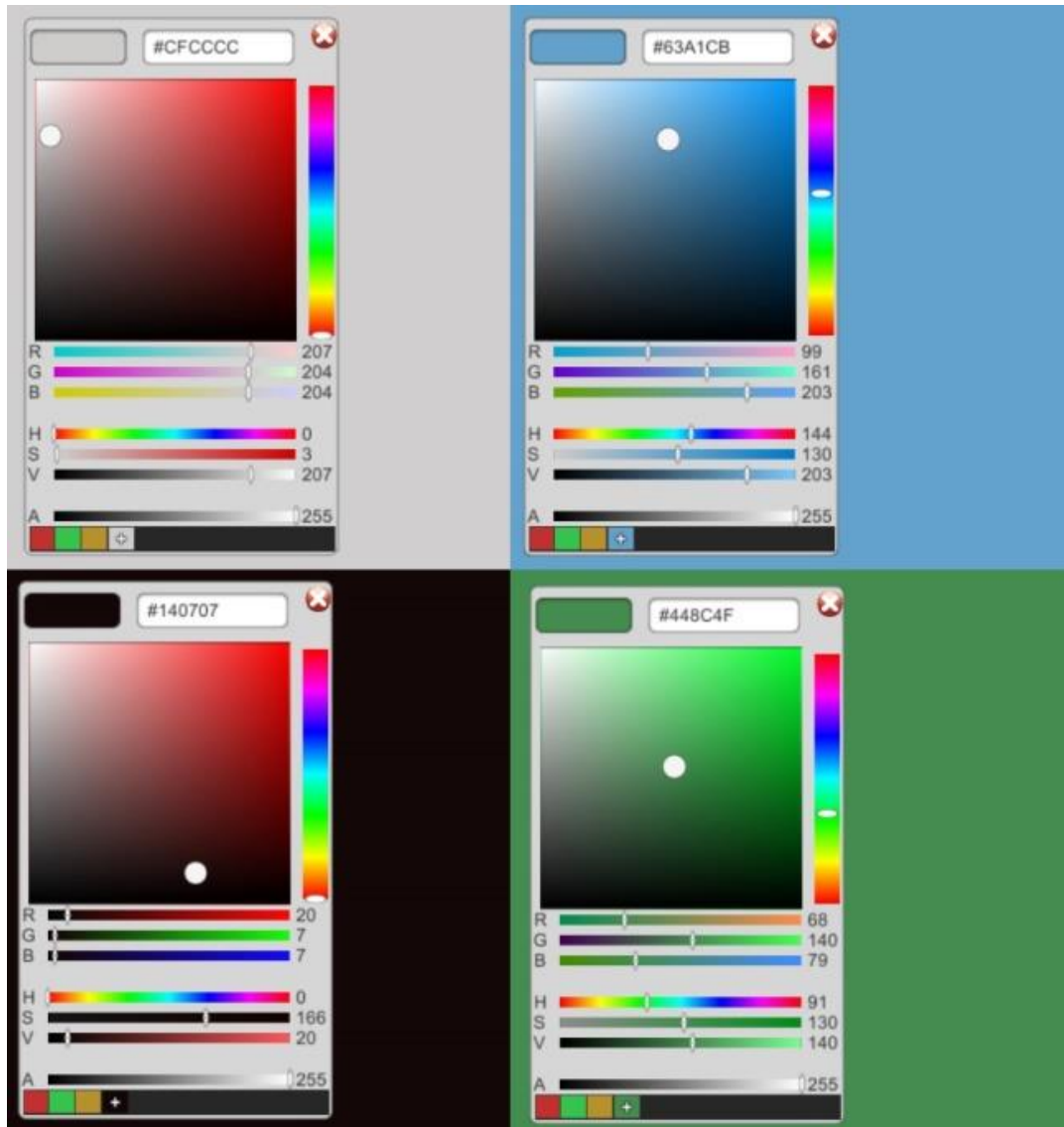


Ilustración 56 Panel de Selección de color, con diferentes variantes de color

7.2.3 Pruebas de Unidad

El cambio de color se aplica correctamente y no se han encontrado problemas al cerrar o abrir este panel, los valores del panel no se modifican.

7.3 Pruebas de Integridad

Tras la realización de pruebas por parte de todo el equipo se ha detectado un funcionamiento anómalo en la interfaz que dificulta la navegación por los paneles. Los paneles de proyectos, opciones, idiomas y selección de color deben ser excluyentes entre sí, es decir, que sólo pueda permanecer uno abierto al mismo tiempo.

Como solución a este problema se han implementado en el script GUI Manager mecanismos para controlar que cuando se requiera abrir un panel se cierre el actual, si hay alguno abierto.

No se han detectado otros errores. El sistema responde correctamente y es fácil de usar.

8. Cierre del proyecto

Con estas tareas se da el proyecto por incluido y se generan los ejecutables que se entregarán junto a esta documentación.

8.1 Generación de los ejecutables

Para la generación de los ejecutables se accederá al panel "File" > "Build Settings". Como ya se configuró la parte relativa a android al comienzo de la iteración 3 el procedimiento a seguir es tan fácil como elegir la plataforma en la lista izquierda y pulsar el botón "Build". La generación del ejecutable para Windows es bastante rápida, unos 15 segundos, mientras que la generación de la apk puede demorarse hasta 10 minutos.

Los ejecutables generados y entregados junto con esta documentación serán Prototipo.exe (junto con su carpeta Prototipo_Data necesaria para su funcionamiento) y HDP Prototipo.apk. También se adjunta el proyecto de prueba TEST-RAKAN y la carpeta "lang" con 2 ficheros de idiomas (español e inglés).

8.6. Evaluación final

Junto con la versiones finales del proyecto se ha entregado al equipo un cuestionario sobre el sistema que deben completar después de testarlo. Se ha dado concedido un semana para realizar tantas pruebas como sea posible y hacer llegar los cuestionarios completados. Esta encuesta puede servir de punto de partida a la hora de crear las encuestas de usuario necesarias para validar el sistema frente al público. Se valorarán unos aspectos del sistema en una escala del 0 al 10, donde 0 significa estar en total desacuerdo con la afirmación y 10 significa estar totalmente de acuerdo con la afirmación.

La [Tabla 9](#) muestra el cuestionario usado y las puntuaciones obtenidas entre los miembros del equipo.

Evaluación Final				
Nº Aspecto	Aspecto	JPC	CC	AG
1	la interfaz de usuario es amigable y sencilla	9	8	8
2	La interacción táctil resulta satisfactoria	8	9	9
3	La interacción con ratón resulta satisfactoria	9	9	9
4	La interacción con teclado resulta satisfactoria	10	10	10
5	El tiempo empleado en abrir un proyecto en Windows es aceptable	10	10	10
6	El tiempo empleado en abrir un proyecto en Android es aceptable	7	6	7
7	El tiempo empleado para cargar un modelo secundario en Windows es aceptable	10	10	10
8	El tiempo empleado para cargar un modelo secundario en Android es aceptable	8	8	7
9	Los puntos de interés son fácilmente identificables	9	8	9
10	Los puntos de interés son fácilmente accesibles	10	10	10
11	Resulta fácil cambiar el idioma del sistema	9	10	10
12	Resulta fácil ajustar la luminosidad de la escena	10	10	10
13	Resulta fácil cambiar el color del fondo	10	10	10

Tabla 9 Cuestionario de Evaluación Final

Porcentajes de satisfacción (%)	
Interfaz/Interacción	90,00
Tiempos de carga Windows	100,00
Tiempos de carga Android	71,67
Puntos de Interés	93,33
Customización	98,89

Tabla 10 Resumen Porcentajes de satisfacción

Esta *Tabla 10* es un resumen porcentual de los resultados obtenidos. Podemos observar cómo las partes de la encuesta relativas a la customización y tiempos de carga en windows obtienen resultados muy cercanos al 100%. En cambio, los tiempos de carga en Android obtienen el porcentaje más bajo con un 71%. El proceso de carga de los modelos 3D hace un uso intensivo momentáneo de los recursos en los dispositivos móviles. Hay que tener en cuenta este hecho a la hora de intentar abrir proyectos con modelos 3D muy complejos o densos en estos dispositivos.

Porcentajes de satisfacción

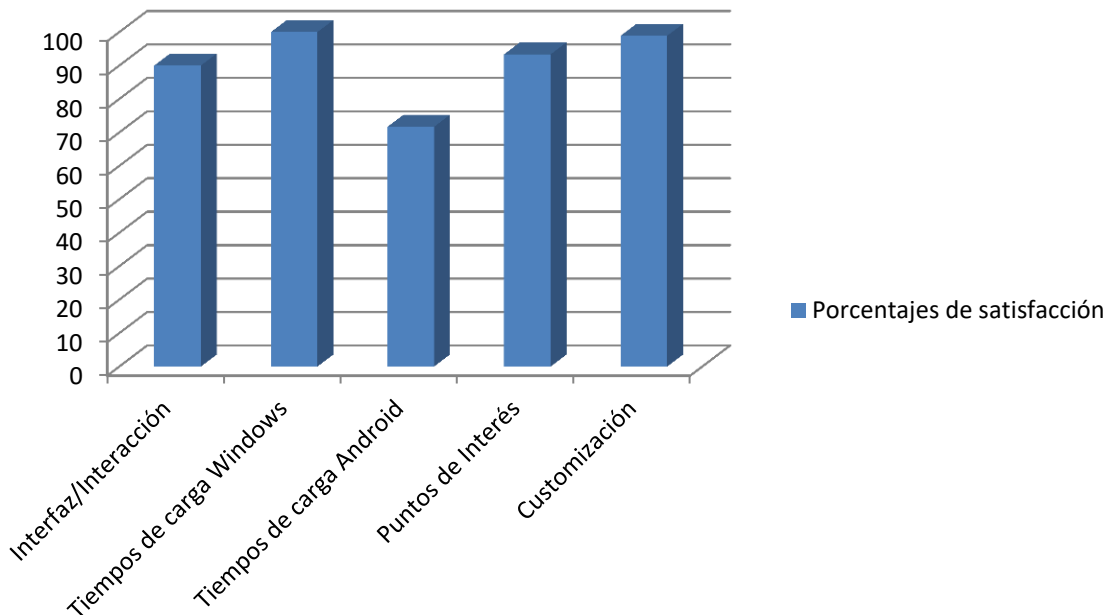


Ilustración 57 Gráfico Porcentajes de satisfacción

8.7. Recursos usados durante el desarrollo

Los recursos externos usados a lo largo del proceso de desarrollo son:

- [iTween](#), plugin gratuito de Unity 3D para realizar pequeñas animaciones dinámicas. Se pueden realizar infinidad de animaciones de una forma muy fácil. Usado para dar un efecto más suave cuando se rota la figura para restablecer su posición inicial o centrar una zona.
- [FileBrowser](#), plugin gratuito de Unity 3D usado para elegir archivos desde disco. Es multiplataforma (Windows y MacOS).
- [HSV Color Picker](#), plugin gratuito de Unity 3D usado para crear un color y, a través de un script propio, modificar el color de fondo de la escena principal.

9. Seguimiento y Control

Como anteriormente se mencionó en el apartado [6.2 Estimación de la Duración](#), el proyecto ha sido dividido en 3 iteraciones, con una duración de 3 semanas cada una. El conjunto de tareas que comprenden el proyecto ha sido dividido en 8 grupos de tareas. Esta organización permite controlar las variaciones de tiempo con un nivel de detalle mucho más exacto (por iteración y por grupo de tareas). Se incluyen 2 subapartados que resumen las comparativas y desviaciones entre tiempo estimado y real. También existe un apartado que resume las horas invertidas por el autor en cada tipo de tarea en concreto.

9.1. Comparativa entre Tiempo Estimado y Real por Iteraciones

RESUMEN TIEMPOS ITERACIONES					
ITERACIONES	TAREAS QUE COMPRENDE	Fecha Inicio	Duración Estimada	Duración Real	Desviación
Iteración 1	0,1,2,3	06/03/2017	43	55	12
Iteración 2	4	27/03/2017	38	45	7
Iteración 3	5,6,7,8	17/04/2017	38	58	20
TOTAL:			119	158	39

Tabla 11 Resumen de Tiempos de las Iteraciones

Comparativa Tiempo Estimado y Tiempo Real de las Iteraciones (en Horas)

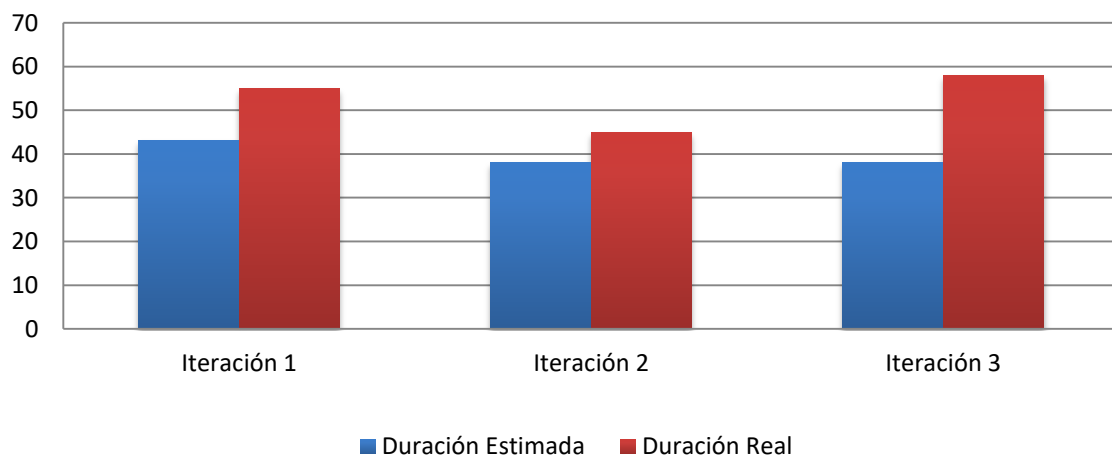


Ilustración 58 Comparativa Tiempos Iteraciones

Diferencia entre Tiempo Estimado y Real por Iteraciones (en Horas)

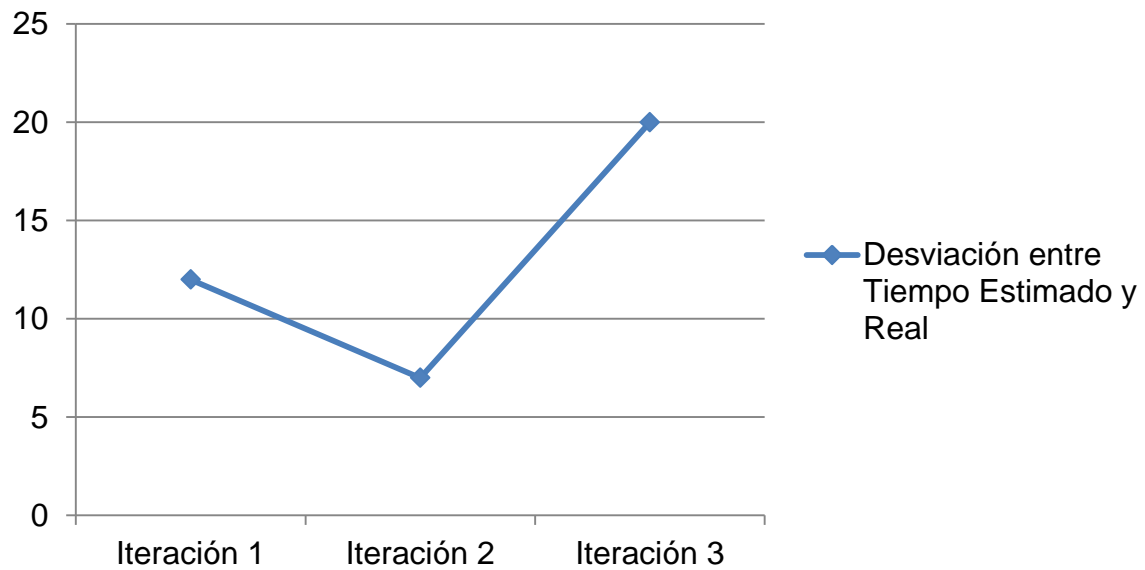


Ilustración 59 Diferencia Tiempos Iteraciones

Como se puede apreciar en la [Tabla 11](#) y en las gráficas ([Ilustración 58](#), [Ilustración 59](#)), todas las iteraciones sufren leves retrasos de tiempo. La iteración 3 es la más afectada con una desviación de 20 horas. En total, el proyecto sufrirá un retraso de 39 horas. En el siguiente punto se mostrará con más detalle la causa de estos retrasos.

9.2. Comparativa entre Tiempo Estimado y Real por Tareas

RESUMEN TIEMPO GRUPOS DE TAREAS			
Grupo de Tareas	Tiempo Estimado (Horas)	Tiempo Real (Horas)	Desviación
0	2	2	0
1	20	28	8
2	10	10	0
3	11	15	4
4	38	45	7
5	15	35	20
6	11	11	0
7	11	11	0
8	1	1	0
TOTAL:	119	158	39

Tabla 12 Resumen de Tiempos de los Grupos de Tareas

Comparativa Tiempo Estimado y Tiempo Real de los Grupos de Tareas (en Horas)

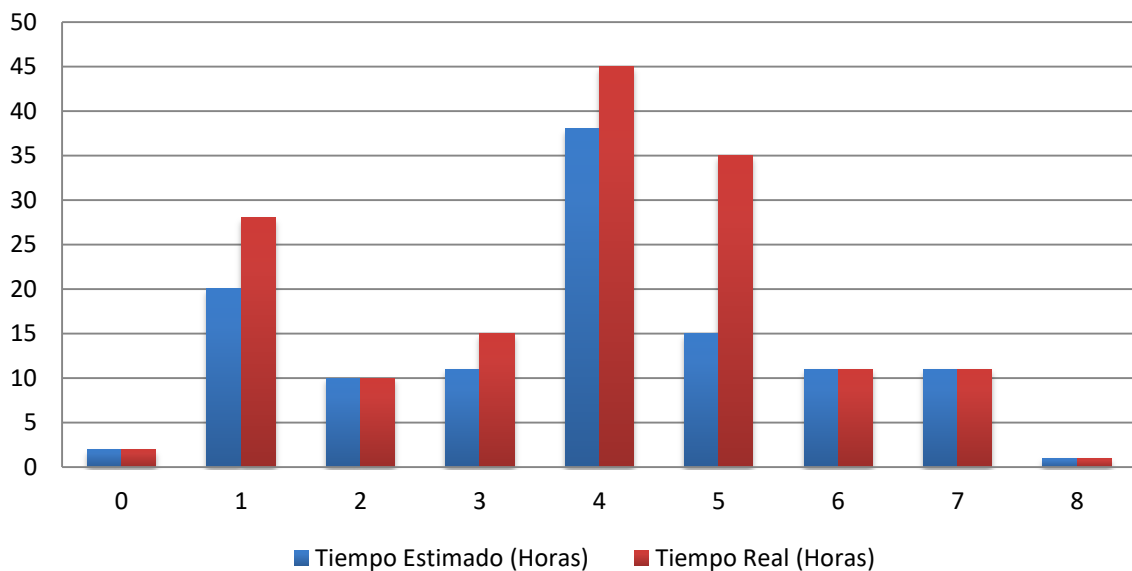


Ilustración 60 Comparativa Tiempos Grupos de Tareas

Diferencia entre Tiempo Estimado y Real por Grupos de Tareas (en Horas)

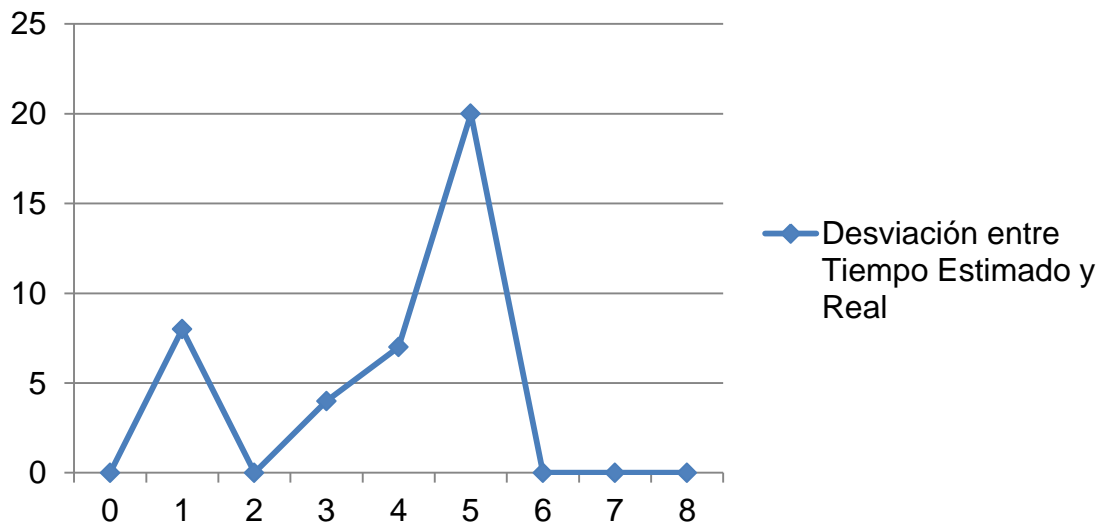


Ilustración 61 Diferencia Tiempos Grupos de Tareas

Esta [Tabla 12](#) y las gráficas ([Ilustración 60](#), [Ilustración 61](#)) ofrecen una visión más detallada de los desfases entre el tiempo estimado y el tiempo real. La mayor diferencia de tiempo se da en el grupo 5 (20 horas), el relativo a la adaptación del sistema a Android. Este retraso está totalmente justificado ya que se tuvieron problemas complejos que hubo que solucionar. Concretamente el problema de las texturas llevó mucho más tiempo del esperado y requirió una gran labor de investigación por parte del equipo.

El siguiente grupo con el desfase más grande se corresponde al grupo 1 de tareas (**Gestión Modelos 3D y Texturas en tiempo real**). Esta diferencia de 8 horas radica en el desconocimiento previo de cómo Unity gestiona internamente los recursos gráficos. El motor no está pensado para cargar modelos 3D y texturas desde disco directamente aunque implementa herramientas para ello, pero no es un proceso trivial.

Los demás grupos retrasados son el 3 (**Proyectos**) y el 4 (**Puntos de Interés**). La variación de tiempo en estos grupos es debido a una estimación optimista por parte del equipo, ya que no se han encontrado problemas a solucionar en las tareas propias de diseño e implementación.

9.3. Coste y Fecha de Finalización

En el apartado *6.4 Estimación de Costes* se estimó un coste total de 732,84 €. Como ya se comentó en la estimación de recursos (*6.3.1 Personas*) la hora de trabajo del desarrollador está valorada en 6,36 € brutos. Un aumento de 39 horas sobre el tiempo estimado del proyecto produce un sobrecoste de 248'04 €, siendo **el total del coste del prototipo de 980,88€**.

A continuación se presentan las fechas estimadas (*Tabla 13*) y reales (*Tabla 14*) de inicio y finalización de las iteraciones. **La finalización del proyecto se retrasará 18 días naturales.**

FECHAS ESTIMADAS DE LAS ITERACIONES		
ITERACIONES	Fecha Inicio Estimada	Fecha Finalización Estimada
Iteración 1	06/03/2017	24/03/2017
Iteración 2	27/03/2017	14/04/2017
Iteración 3	17/04/2017	05/05/2017

Tabla 13 Fechas Estimadas de las Iteraciones

FECHAS REALES DE LAS ITERACIONES		
ITERACIONES	Fecha Inicio Real	Fecha Finalización Real
Iteración 1	06/03/2017	29/03/2017
Iteración 2	30/03/2017	20/04/2017
Iteración 3	21/04/2017	23/05/2017

Tabla 14 Fechas Reales de las Iteraciones

9.4. Horas invertidas por tipo de Tarea

Horas invertidas por tipo de Tarea

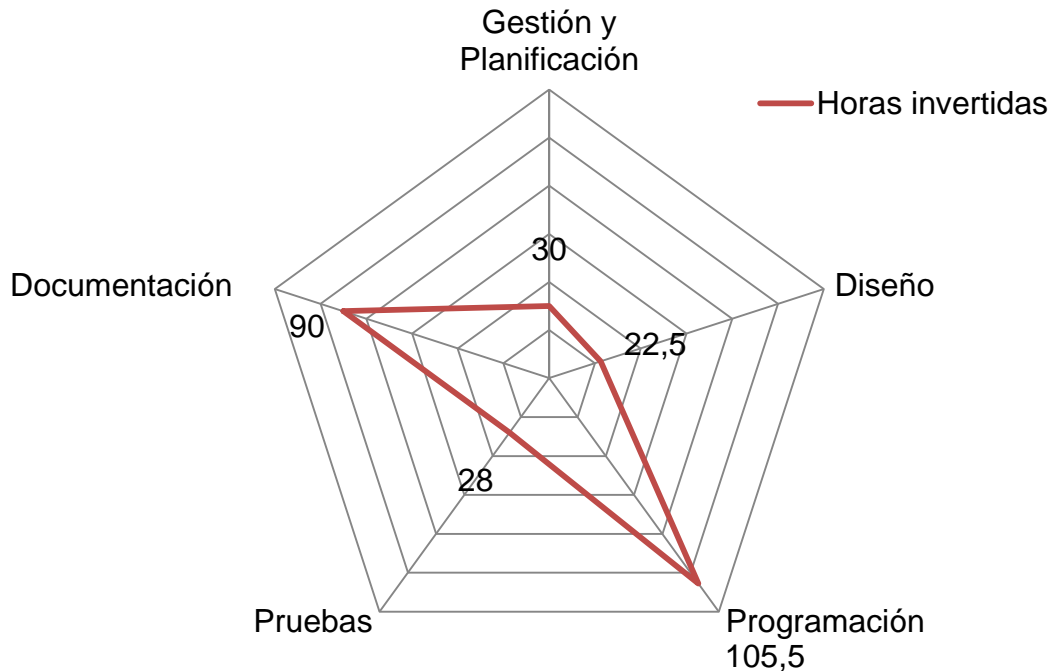


Ilustración 62 Horas invertidas por tipo de Tarea

Con un total de **276 horas**, el gráfico anterior (*Ilustración 62*) muestra la distribución de tiempo invertido durante todo el proyecto. Las realización de las tareas de las iteraciones (Programación, Modelado 3D, Diseño 2D, Diseño Mecánicas y Pruebas) hacen un total de 158 horas, por lo que el 42,75 % del tiempo total ha sido dedicado a tareas de Gestión, Planificación y Control del proyecto y a la elaboración de esta documentación.

10. Finalización del Proyecto

Al final del proyecto se debe generar un informe dirigido al cliente y otro dirigido al equipo de desarrollo. Como el desarrollo ha sido por y para la empresa no existe una figura de cliente como tal. Este hecho no exime de la generación y entrega de estos informes. Para la empresa, el informe constará de los siguientes apartados de este documento:

- *4 Análisis de requisitos*
- *5 Elección de Tecnologías y Herramientas*
- *6 Planificación*
- *7 Diseño Estructural*
- *8 Ejecución*
- *9 Seguimiento y Control*
- *14 Anexo I: Uso de la Herramienta*
- *15 Anexo II: Manual de creación de proyectos*

El siguiente apartado está dirigido al equipo de desarrollo como forma de retroalimentación para una mejora en la ejecución de similares proyectos en el futuro.

10.1. Informe para el Equipo de Desarrollo

Técnicas utilizadas para conseguir los objetivos propuestos

- Se ha usado una metodología iterativa incremental, teniendo funcionalidades concretas acabadas en una fase temprana del proyecto.
- Se ha seguido exitosamente el típico esquema de actuación de tres pasos diseño, implementación y prueba.
- Todas las funcionalidades han sido probadas individualmente (en su fase de pruebas correspondiente), antes de realizar pruebas de integración en el sistema completo.
- Se han realizado pruebas de usuario con personas ajenas al equipo de desarrollo para recoger opiniones y críticas objetivas.
- Todos los miembros de la empresa han participado activamente en las fases de diseño y pruebas.
- Se han aplicado patrones tales como el patrón “Composite” (Componentes) en todo el proyecto y el patrón arquitectónico Modelo-Vista-Controlador (*7.1.2 GUI Manager*).
- Cada 2 días se han realizado copias de seguridad en un disco duro externo y se ha usado GIT para el control de versiones.

- Uso intensivo de la documentación de [Unity3D](#).
- Se han usado activamente foros como [stackoverflow](#) y el [foro oficial](#) de Unity para la consulta de problemas puntuales o situaciones similares a las superadas en este proyecto.

Fortalezas manifestadas durante el desarrollo

- Capacidad de resolución de problemas ante nuevas situaciones, búsqueda y estudio de las soluciones más viables.
- Rápida inmersión en nuevos campos de desarrollo, como el tratamiento de gráficos 3D y texturas.
- Constancia en el proceso de desarrollo.
- Correcta aplicación del modelo de ciclo de vida de la gestión del proyecto.
- Estructuración del flujo de trabajo válida en la ejecución del proyecto.
- Autoaprendizaje y autonomía.

Debilidades manifestadas durante el desarrollo

- Estimación de la duración del proyecto algo optimista.

Recomendaciones al equipo

Ante un desarrollo similar, sería beneficioso dar al equipo un curso avanzado sobre el framework a usar. Se acelerará el proceso de desarrollo y se podrán realizar estimaciones más certeras del trabajo (en tiempo y esfuerzo) que puede suponer una tarea.

UNIDAD III: RESULTADOS Y CONCLUSIONES

“La calidad nunca es un accidente, siempre es el resultado de un esfuerzo de la inteligencia.”

John Ruskin

11. Exposiciones

11.1. La obra invitada: Inmaculada Napolitana



la obra invitada

INMACULADA

ESCUELA NAPOLITANA, CA. 1695

3-13/octubre/2017

Espacio "Obra invitada"
(Antigua Escuela de Magisterio)

Lunes a viernes de 10.00 a 13.30
y de 17.00 a 20.30 horas.
Cerrado festivos y no lectivos.

 Universidad
de Jaén

**UJa.
Cultura**

 EXCMO. CABILDO DE LA
S. I. CATEDRAL DE JAÉN

Más información:
WEB: www.uja.es/cultura
FACEBOOK: [uja.cultura](https://www.facebook.com/uja.cultura)
TWITTER: [biua.cultura](https://twitter.com/biua.cultura)
INSTAGRAM: [uja.cultura](https://www.instagram.com/uja.cultura)



Ilustración 63 Cartel exposición "La obra invitada: Inmaculada"



Ilustración 64 Edificio Antigua Magisterio de la Universidad de Jaén, lugar de la exposición



Ilustración 65 Panfletos informativos del evento

Esta exposición ha servido para presentar la herramienta al público desde el 3 de octubre de 2017 hasta noviembre. Como ya se comentó en el apartado [3.3 Modelado: “La Inmaculada Napolitana”](#), la empresa realizó el modelado de la Inmaculada Napolitana antes de su restauración. La universidad de Jaén prestó el ordenador con pantalla táctil para ejecutar el sistema. Se creó un proyecto con el modelo 3D en alta resolución, el modelo en color y otros dos modelos secundarios (uno en color y otro con luz ultravioleta) de las patas de la obra. También se introdujo información y fotos facilitadas por el equipo de 4D Geoservices y el restaurador de la catedral de Jaén.



Ilustración 66 Visitante manipulando el Modelo 3D



Ilustración 67 Día de la Inauguración

11.2. Catedral de Jaén

Desde enero de 2018 hasta la actualidad (18/06/2018) se puede disfrutar del sistema junto con la talla original en la catedral de Jaén.

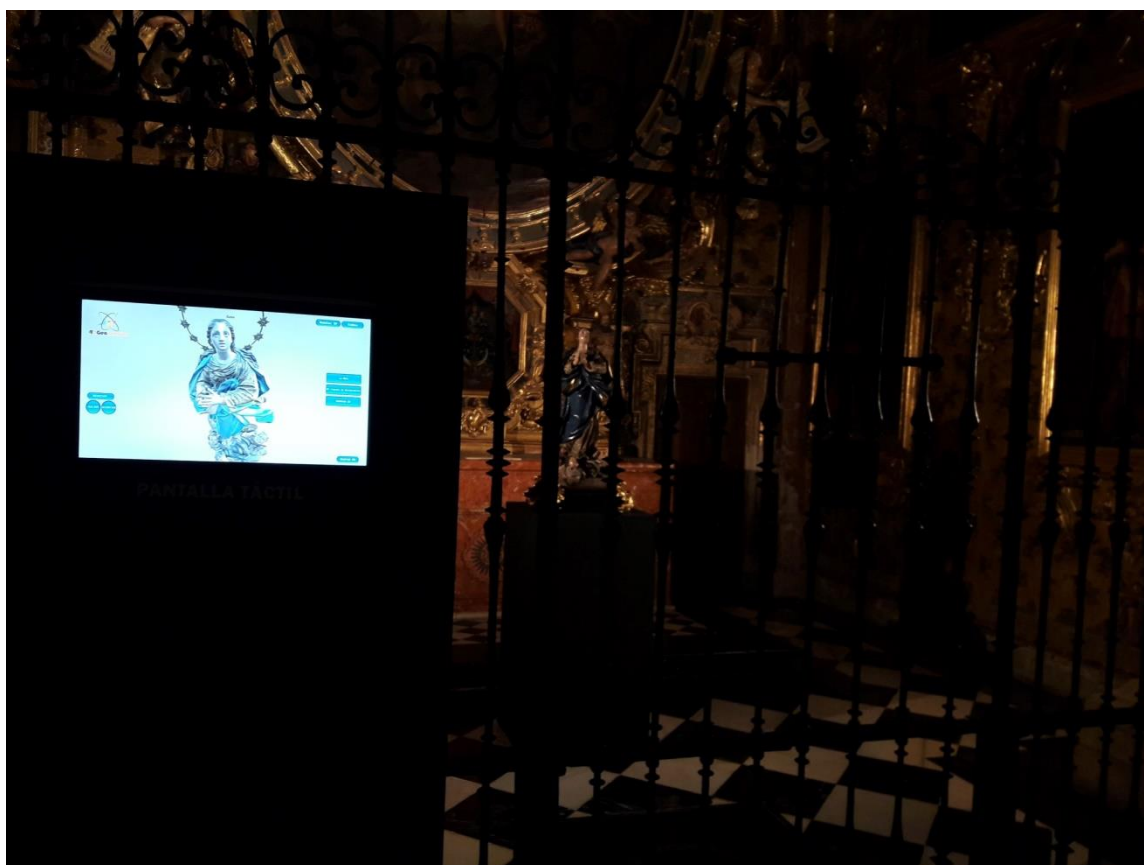


Ilustración 68 La Herramienta en la Catedral de Jaén 1

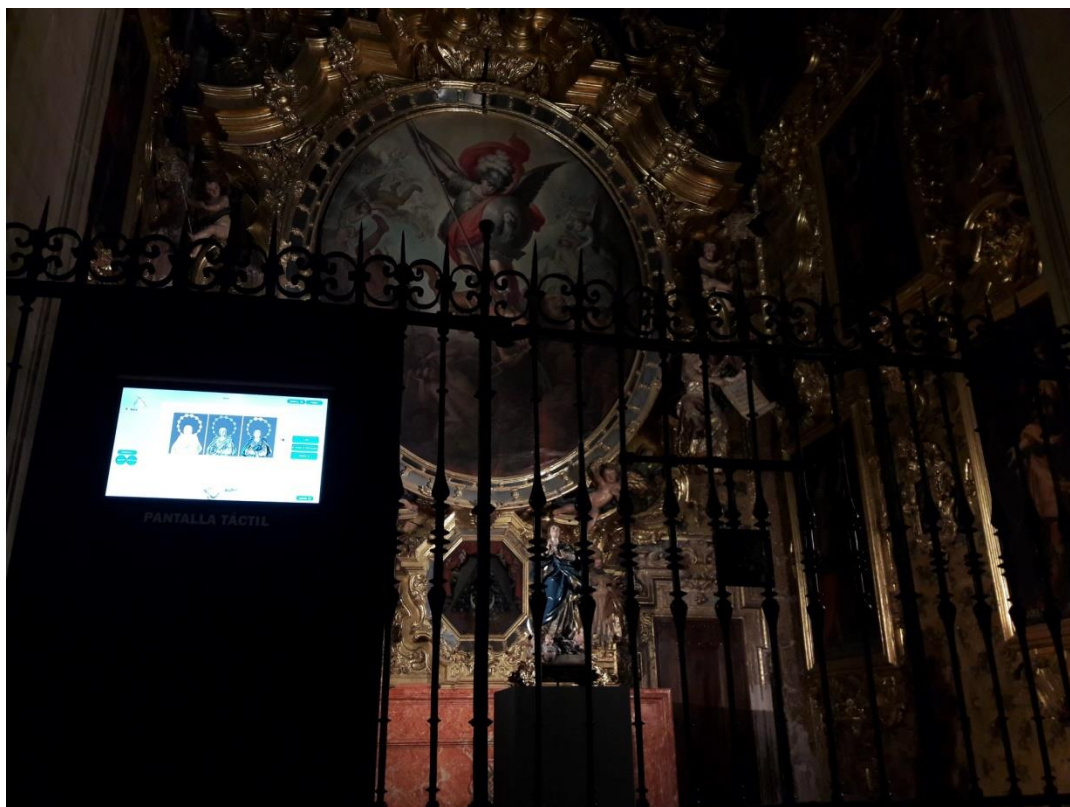


Ilustración 69 La Herramienta en la Catedral de Jaén 2



Ilustración 70 La Herramienta en la Catedral de Jaén 3



Ilustración 71 La Herramienta en la Catedral de Jaén 4



Ilustración 72 La Inmaculada Napolitana restaurada en la Catedral de Jaén

12. Futuras Mejoras

Actualmente no se contempla una futura actualización del sistema ya que existen otros proyectos más prioritarios. La lista de futuras mejoras, derivadas del uso del sistema y su exposición al público son:

- Inclusión de audios y videos en los puntos de interés.
- Una interfaz más atractiva visualmente.
- Que los modelos secundarios posean sus propios puntos de interés.
- Que los proyectos se descarguen de remotamente desde un repositorio central.
- Poder cargar un proyecto existente en cualquier parte del sistema de directorios.
- Extender la herramienta, crear la versión de conservación de patrimonio, para cumplir el doble objetivo de la empresa ([1.4 Motivación](#)).

13. Conclusiones

Desde el primer día que se me habló sobre este proyecto he estado muy ilusionado con él. Creo que es un sistema muy interesante, lleno de posibilidades si se dirige correctamente a clientes potenciales. Dar a los usuarios el poder de manipular los modelos 3D y ofrecerles información en multitud de formatos puede cambiar la forma de visitar museos y exposiciones.

Este proyecto ha supuesto un reto a nivel técnico, enfrentándome exitosamente a las complicaciones que han surgido a lo largo del proceso. Estas complicaciones, junto con una estimación algo optimista han creado el desfase de tiempo (y costes) presente en el apartado [9 Seguimiento y Control](#). Si bien este factor no ha sido tan crítico al ser un desarrollo propio para la empresa me deja un sabor agrisado al cierre del proyecto. Como buena práctica futura realizaré estudios iniciales de las herramientas a usar en un proyecto de forma más exhaustiva.

Durante el proceso de desarrollo he aprendido sobre modelado y escaneo 3D, una gestión más avanzada de geometrías y texturas y características propias del motor de videojuegos usado (entradas táctiles por ejemplo). Estos conocimientos pueden ser aplicados en nuevos proyectos, por lo que el conocimiento extraído de éste desarrollo es potencialmente muy valioso. También he ganado experiencia en el uso de metodologías iterativas, en las que las tareas de diseño, implementación y pruebas se intercalan en iteraciones más dinámicas, controlables y efectivas.

Unity es un framework muy potente, con infinidad de posibilidades, no sólo limitado al ámbito del entretenimiento. Un claro ejemplo de ello es este proyecto. He descubierto funcionalidades y comportamientos de Unity desconocidos para mí hasta ahora, y empecé el proyecto una importante experiencia previa. Creo que cualquier personal técnico debería ser formado en motores de uso general como éste, las posibilidades y potencia que ofrecen son enormes.

Aunque actualmente otros proyectos consumen mi jornada laboral con la empresa, me gustaría retomar el proyecto en un futuro para aplicar las mejoras del punto [12 Futuras Mejoras](#) y tener un sistema más potente y completo.

UNIDAD IV: ANEXOS

“Recuerda: no eres torpe, no importa lo que digan esos libros. Los torpes de verdad son gente que, creyéndose expertos técnicos, no podrían diseñar hardware y software manejable por usuarios normales aunque la vida les fuera en ello.”

Walter Mossberg

14. Anexo I: Uso de la Herramienta

Interfaz



Ilustración 73 Interfaz de Usuario

La interfaz de usuario, *Ilustración 73*, se divide en

- **Panel de navegación izquierdo:** Contiene los controles de movimiento del modelo 3D.
- **Barra de menú superior:** Da acceso a los paneles de selección de proyecto, opciones, modelos secundarios, cambia al modo creación de puntos de interés y guarda los cambios realizados a éstos puntos.
- **Panel de puntos de interés (Derecha):** Contiene una lista de accesos rápidos a los puntos de interés definidos en el proyecto.

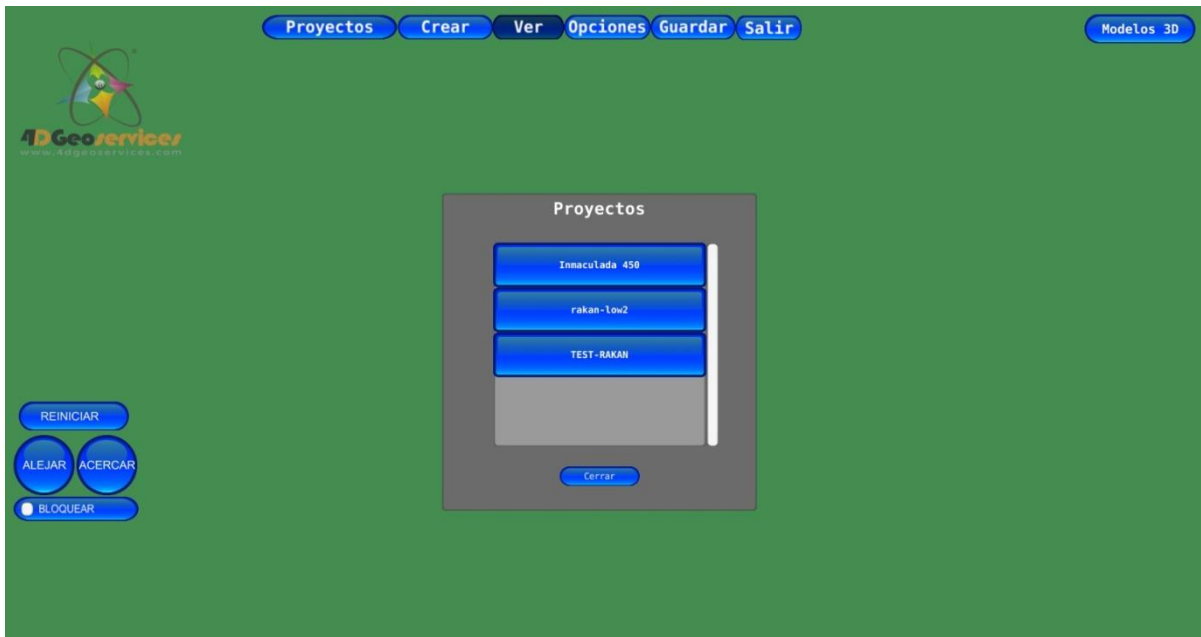


Ilustración 74 Panel Selección de Proyectos

El panel “Proyectos”, *Ilustración 74*, permite cargar cualquier proyecto existente en el sistema.

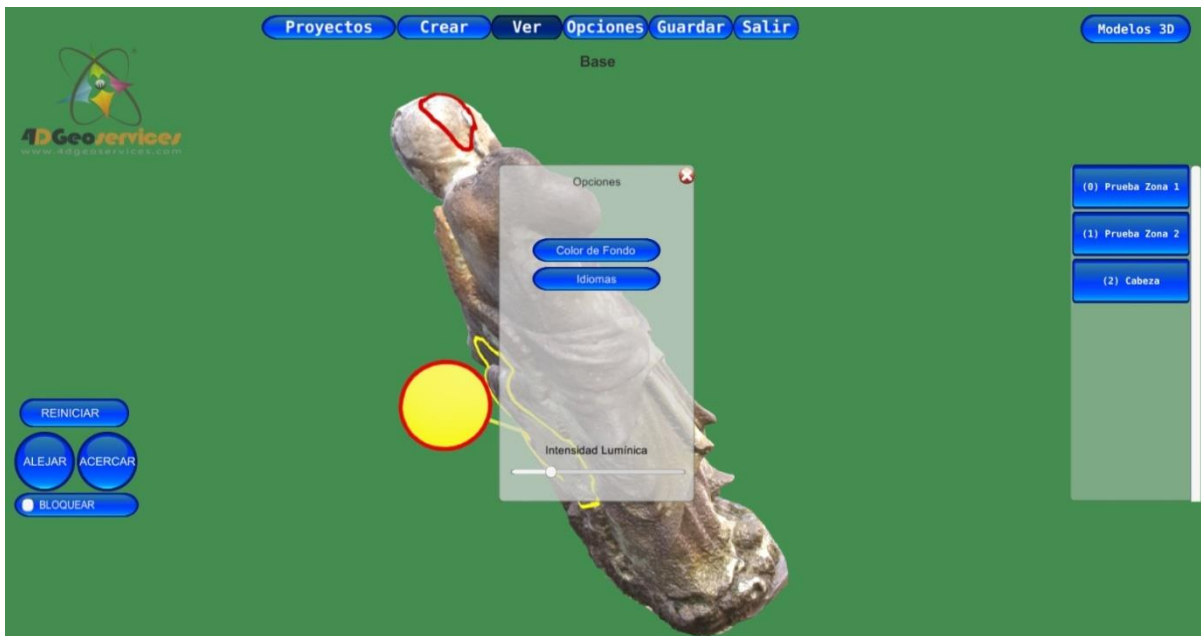


Ilustración 75 Panel de Opciones

El panel de opciones, *Ilustración 75*, permite cambiar el idioma, ajustar la intensidad lumínica y cambiar el color del fondo de escena.



Ilustración 76 Panel selección modelos secundarios

El botón Modelos 3D junto con su panel de selección (*Ilustración 76*) permite cargar los modelos secundarios de un proyecto abierto.

Creando un Punto



Ilustración 77 Creando un Punto de Interés

A través de un botón (“Crear”) en la interfaz se habilita la creación de puntos de interés sobre el modelo. Se recomienda usar la función bloquear (freeze) de la cámara para que, al arrastrar el ratón sobre la figura ésta no se mueva. Como se puede apreciar en la *Ilustración 77*, se van creando pequeños puntos rojos sobre el modelo. Al soltar el ratón podremos completar este punto

de interés con información o descartarlo a través del panel que se presenta en la siguiente *Ilustración 78*.



El panel, titulado "Área de Interés", está diseñado para la gestión de datos de un punto de interés. Presenta un encabezado con el título "Área de Interés" y el campo "ID". A continuación, se encuentran los campos "Nombre" y "Descripción", ambos con un formato de tabla de 2x2. El campo "Descripción" ocupa un espacio mayor que el de "Nombre". En la parte inferior del panel, se encuentran tres botones de acción: "Guardar", "Borrar" y "Cerrar".

Área de Interés			
ID			
Id			
NOMBRE			
Nombre			
DESCRIPCIÓN			
Descripción			

Guardar Borrar Cerrar

Ilustración 78 Panel para completar información de un punto de interés

Añadiendo imágenes

Una vez creado el punto, podemos agregar imágenes que serán mostradas cuando se consulte el punto.

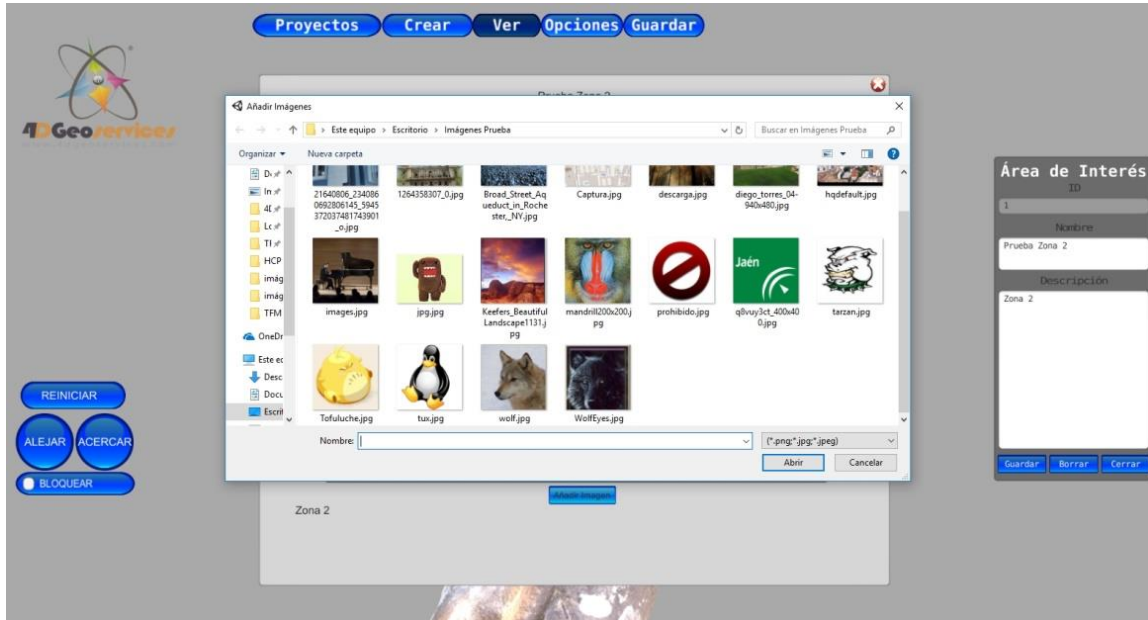


Ilustración 79 Añadiendo Imágenes a un punto de interés

Consultando Puntos

Podemos consultar un punto de interés en concreto seleccionando su área o en la lista de accesos directos de la derecha.

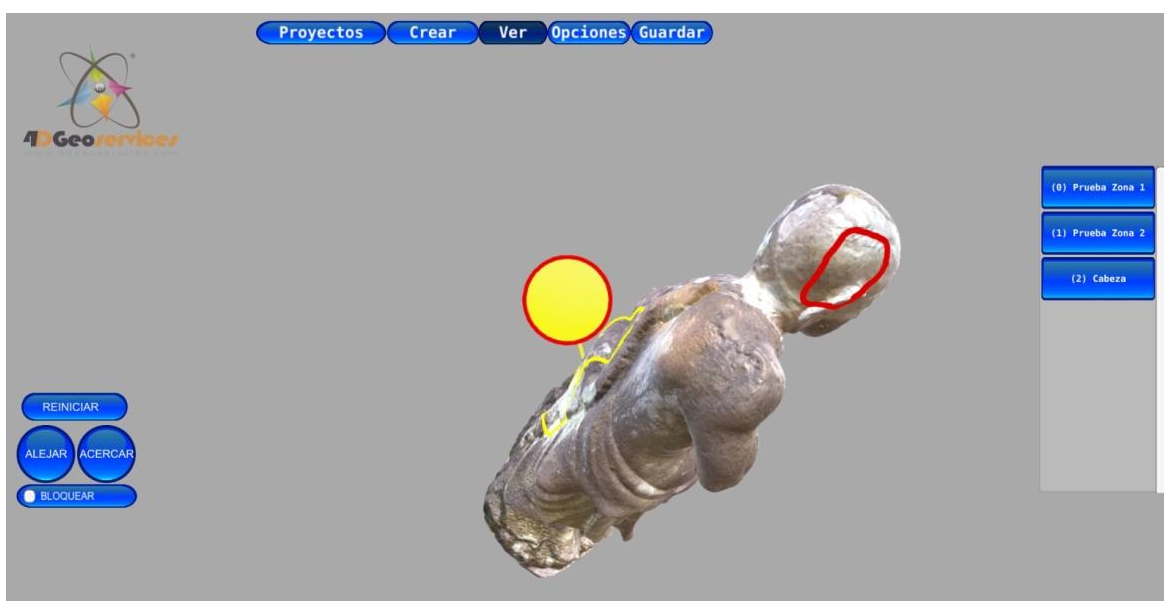


Ilustración 80 Punto de interés seleccionado, con un marcador

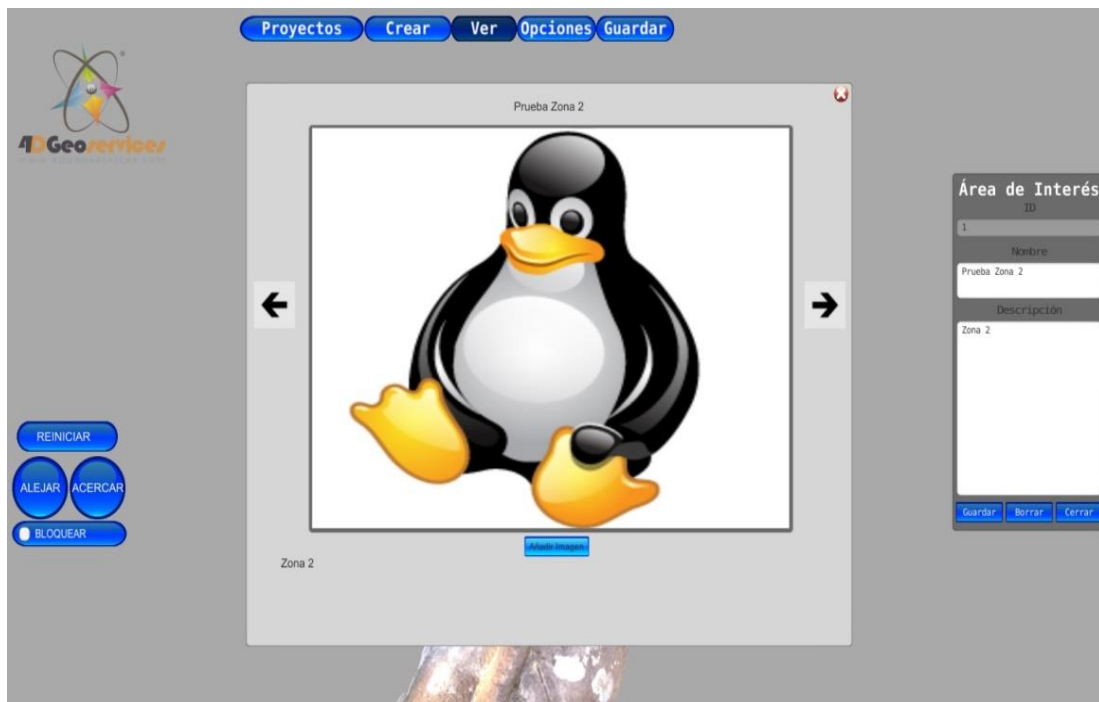


Ilustración 81 Información asociada a un punto de interés

Una vez seleccionado un punto se abrirán los dos paneles de la imagen anterior, el panel con la información textual de la zona y el panel de galería de imágenes agregadas a ese punto.

Controles de Navegación

Ratón

- Scroll Rueda Ratón hacia arriba: Aumentar zoom al modelo.
- Scroll Rueda Ratón hacia abajo: alejar zoom del modelo.
- Arrastrar ratón en eje X positivo: Rotar modelo a la derecha.
- Arrastrar ratón en eje X negativo: Rotar modelo a la izquierda.
- Arrastrar ratón en eje Y positivo: Rotar modelo hacia arriba.
- Arrastrar ratón en eje Y negativo: Rotar modelo hacia abajo.

Teclado

- Tecla +: Aumentar zoom al modelo.
- Tecla -: alejar zoom del modelo.
- Tecla Dirección Derecha: Rotar modelo a la derecha.
- Tecla Dirección Izquierda: Rotar modelo a la izquierda.
- Tecla Dirección Arriba: Rotar modelo hacia arriba.
- Tecla Dirección Abajo: Rotar modelo hacia abajo.

Pantalla Táctil

- Arrastrar dedo en eje X positivo: Rotar modelo a la derecha.
- Arrastrar dedo en eje X negativo: Rotar modelo a la izquierda.
- Arrastrar dedo en eje Y positivo: Rotar modelo hacia arriba.
- Arrastrar dedo en eje Y negativo: Rotar modelo hacia abajo.

Panel de Navegación



Ilustración 82 Panel de Navegación

Archivos e Instalación

La versión Android del sistema es una apk que puede instalarse de forma normal activando la instalación software de “Orígenes Desconocidos” en el dispositivo.

La versión de Windows consta de los siguientes archivos.

- **HCP.exe:** El ejecutable de la aplicación.
- **HCP_Data:** Los archivos necesarios para que el ejecutable funcione. El .exe y esta carpeta deben estar siempre en el mismo directorio o la aplicación no funcionará.

La estructura básica de un proyecto será una carpeta que contenga toda esta información:

- **Archivo .dat** del modelo principal, la geometría.
- **Archivo .cfg** con la configuración del modelo principal.
- **Carpeta textures**, con las texturas del modelo principal.
- **Archivos .ctx en la carpeta textures**, texturas comprimidas.
- **Carpeta zones**, necesaria para guardar los puntos de interés.
- **Archivo .zones en carpeta zones** con los puntos de interés del modelo principal.
- **Carpetas en carpeta zones** para las imágenes de los puntos de interés.
- **Carpeta submeshes**, con los modelos secundarios (y sus texturas).

Los proyectos deben encontrarse en un directorio específico del sistema. En sistemas windows este directorio se encuentra en “C:\Users\<USUARIO>\AppData\LocalLow\<EMPRESA>\<NOMBRE DEL PROYECTO>”. Donde USUARIO es el nombre del usuario de Windows, EMPRESA es el nombre de la empresa (parámetro personalizable en el proyecto Unity, por defecto es DefaultCompany), y NOMBRE DEL PROYECTO es, en este caso concreto HCP DESKTOP.

En sistemas Android se usa la carpeta “files” dentro del propio directorio de la aplicación.

En estos directorios también debe incluirse la carpeta “lang” suministrada con el sistema, ya que contiene los ficheros de idiomas externos a éste.

15. Anexo II: Manual de creación de proyectos

Este es el manual entregado al personal de la empresa para la creación de proyectos. No ha sufrido ninguna modificación respecto al original. El siguiente software necesario se entregará junto con esta documentación (menos Unity 3D que se puede descargar gratuitamente desde su [web oficial](#)).

Software necesario:

- Un modelo 3D (.obj) y su textura (.jpg). La textura debe estar en un solo archivo.
- Tener instalado Unity 3D (mínimo la versión **2017.2.0f3**).
- El proyecto HCP DESKTOP (o el paquete HCP DESKTOP.unitypackage).

Pasos necesarios:

- 1) Abrir el proyecto con Unity 3D. Si se tiene el paquete, crear un nuevo proyecto 3D vacío e importar el paquete en el menú Assets/import package/custom package. Este paso puede llevar unos minutos si se usa por primera vez una versión más moderna, ya que recompilará los scripts y reimportará los assets para que todo funcione correctamente.
- 2) Arrastrar el modelo .obj y la textura a la carpeta Assets/Import (tab "Project"). Dependiendo del tamaño de estos archivos puede tardar unos minutos. En este paso se realiza la importación del modelo, Unity aplica una serie de optimizaciones que mejorarán el rendimiento del programa. Una vez finalizado este paso, veremos el modelo, la textura y una carpeta del materiales que Unity crea para el modelo.

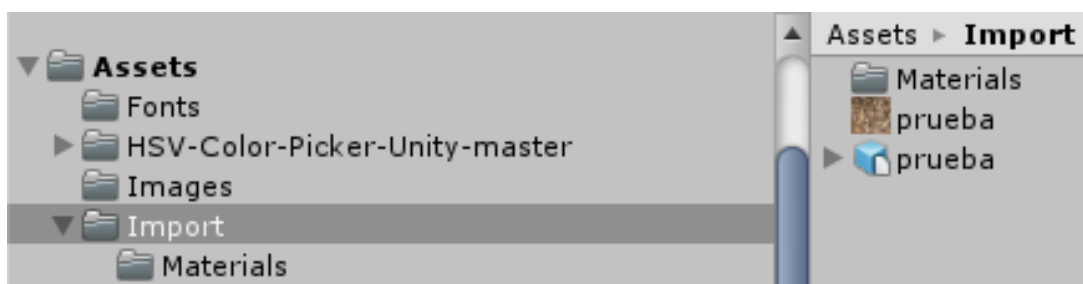


Ilustración 83 Carpeta "Import" para la importación del Modelo

- 3) Seleccionamos la textura y en “Advanced” la configuramos de la siguiente forma:
 - a) **Activar** Read/Write Enabled
 - b) **Desactivar** Generate Mip Maps
 - c) Pulsar **Apply**.

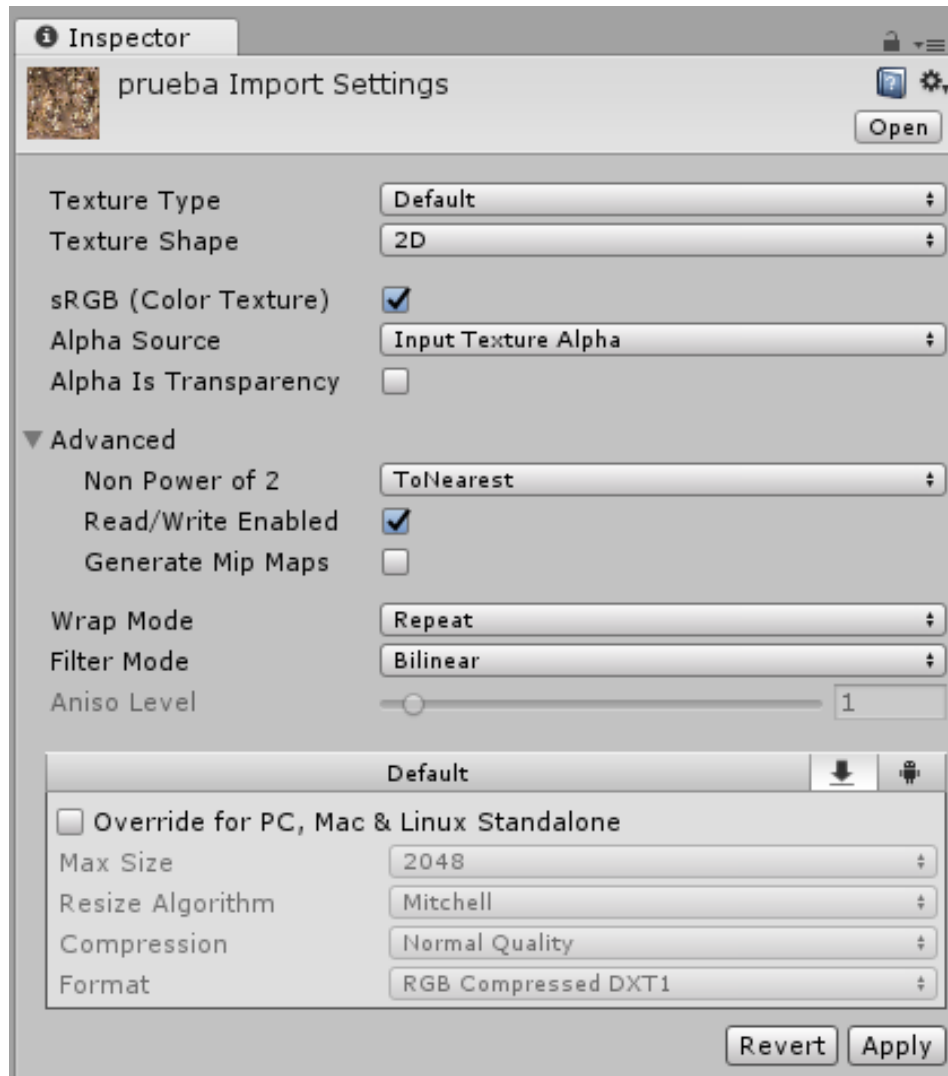


Ilustración 84 Configuración de importación de Textura

- 4) En la carpeta Materials (creada automáticamente) seleccionamos el material defaultMat y arrastramos la textura al campo “Albedo”.

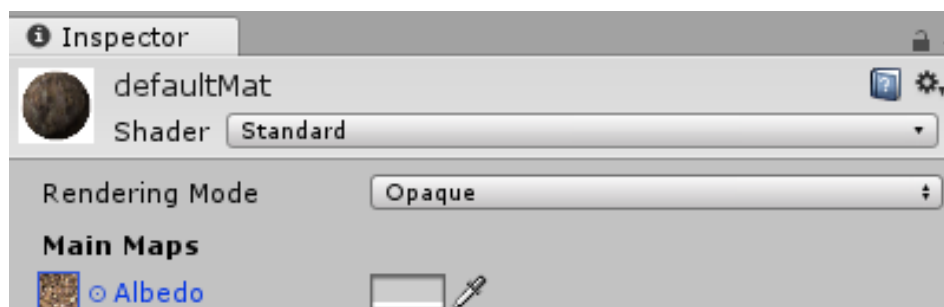


Ilustración 85 Asociación de Textura al material

- 5) Arrastramos el modelo (prueba en este caso) a la jerarquía de la escena (tab "Hierarchy") de modo que sea hijo de Origin pero no hijo de ModelRoot.

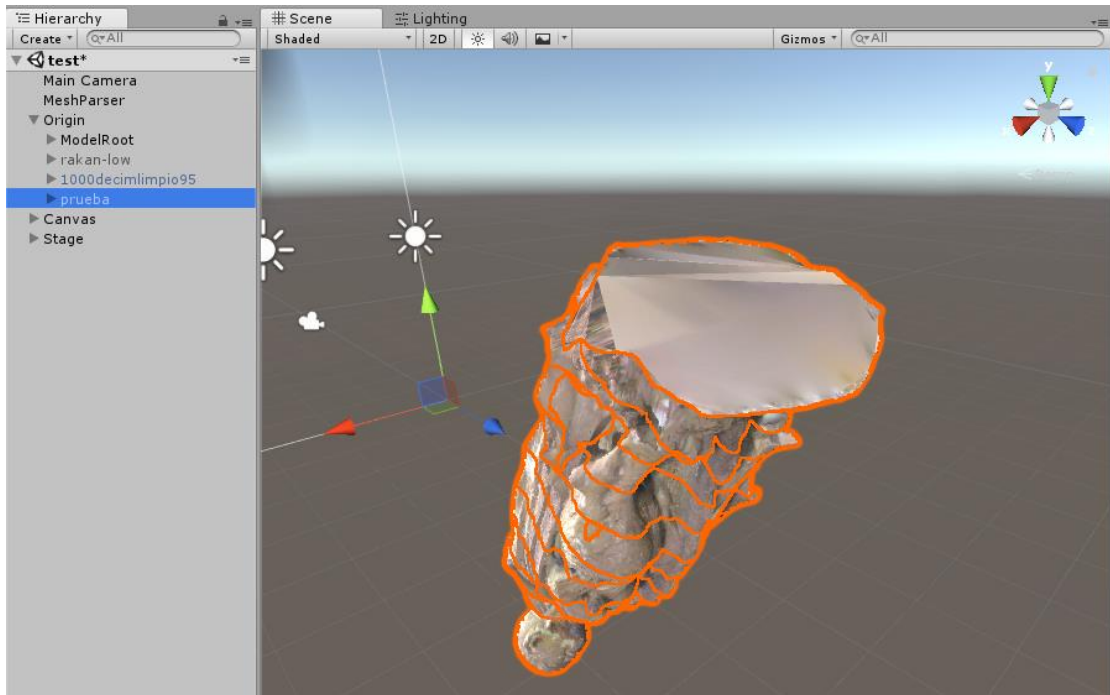


Ilustración 86 Modelo de Pruebas siendo posicionado en la escena

- 6) Seleccionar el modelo en la jerarquía y con las herramientas de **traslación** (W), **rotación** (E) y **escalado** (R) ajustar la posición inicial del modelo (se recomienda ir controlando el resultado en el tab Game).
 - a) Como hijo de ModelRoot existe un Cubo (el centro) que puede activarse para que sirva de guía.

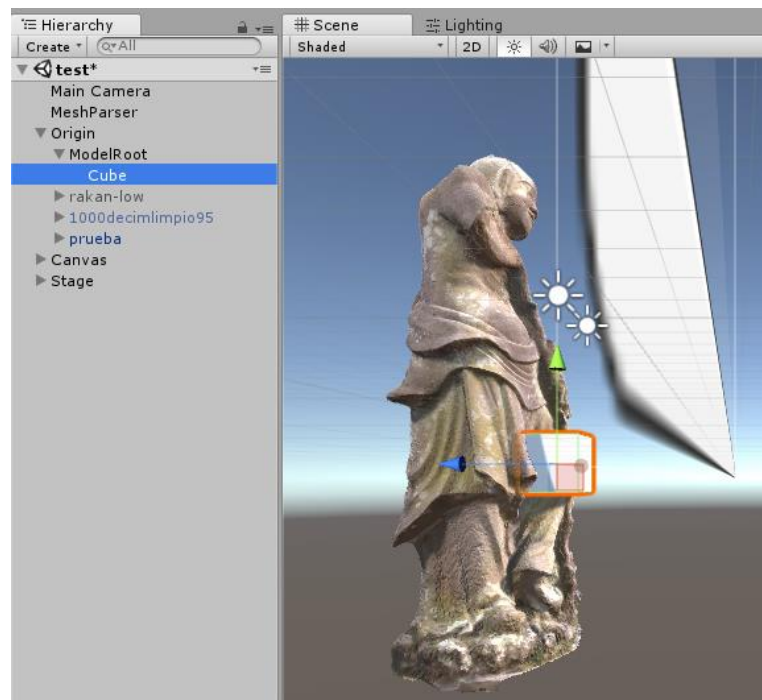


Ilustración 87 Cubo en el Origin que sirve como guía

- b) Pulsando Play en cualquier momento podemos comprobar la rotación y la posición del modelo. Cualquier cambio efectuado mientras se está en "Play" será revertido al finalizar este.

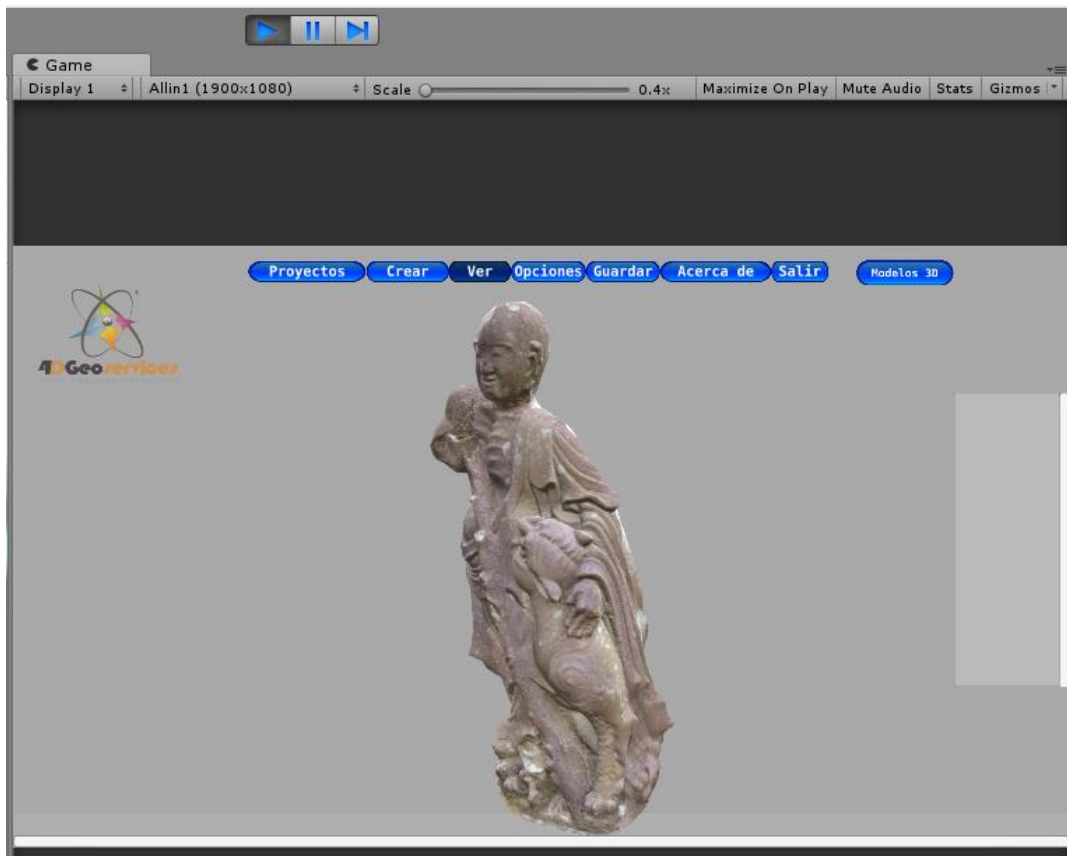


Ilustración 88 Al pulsar "Play" comprobamos la correcta posición

- 7) El objeto “MeshParser” de la escena (tab Hierarchy) contiene un script llamado “Complex Mesh Serializer”.

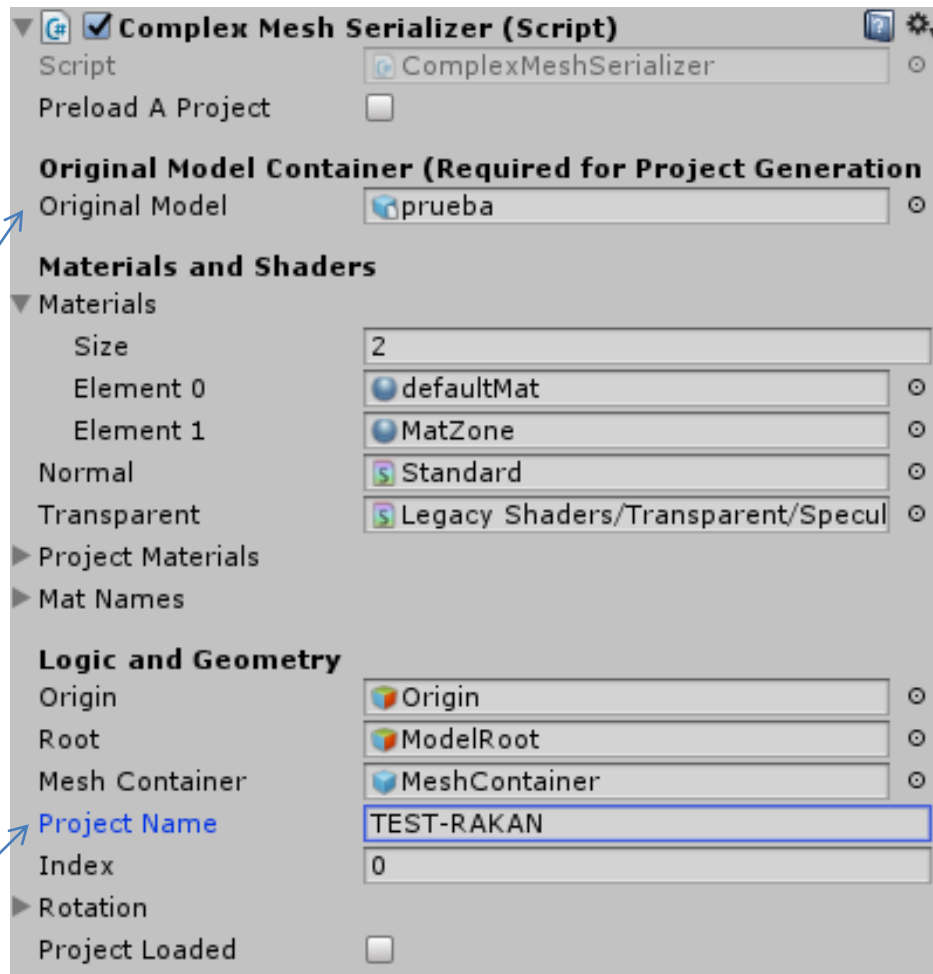


Ilustración 89 Script Complex Mesh Serializer

- En Original Model arrastrar el modelo (**OJO**: el modelo de la escena, el que anteriormente ajustamos, no el de la carpeta Import).
- En Project Name poner un nombre al gusto (comprobar que no exista un proyecto con el mismo nombre o se sobrescribirá).
- Pulsar Play, activar la tab Console, terminar de ajustar el modelo si es necesario y cuando todo esté correcto pulsar la tecla “C”.

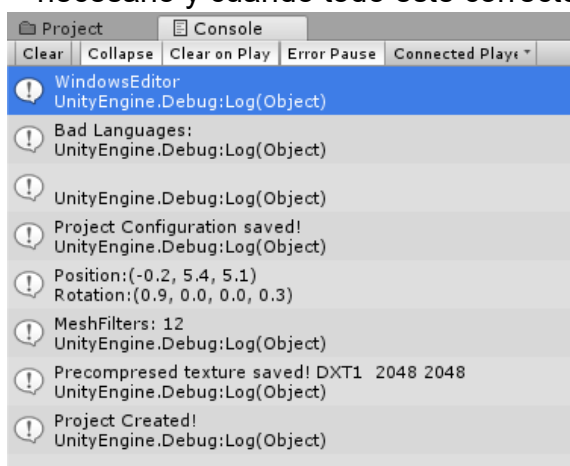


Ilustración 90 Log del sistema al crear el proyecto

Si todo ha salido correctamente se verá un log parecido a este.

- 8) Pulsar Stop, borrar el modelo de la tab "Hierarchy" y comprobar que efectivamente el proyecto ha sido creado y es accesible.

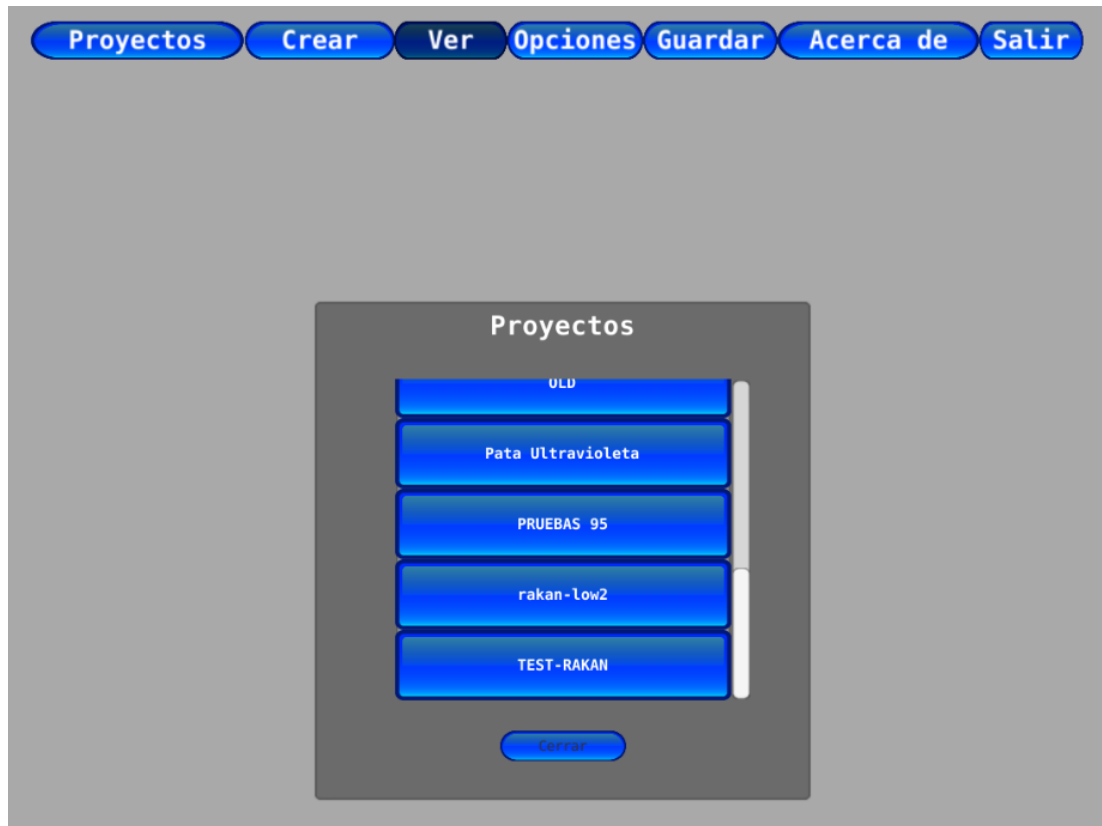


Ilustración 91 Lista de proyectos existentes en el sistema

¿Cómo añadir un modelo secundario a un proyecto?

Tenemos 2 proyectos diferentes: TEST-RAKAN e Inmaculada 450.

Para que el modelo base de Inmaculada 450 sea añadido como secundario en TEST-RAKAN haremos lo siguiente:

- a) Entrar en la carpeta del proyecto Inmaculada 450 y copiar la carpeta textures, el .cfg y el .dat a la carpeta submeshes del proyecto TEST-RAKAN, quedando así.

> AppData > LocalLow > DefaultCompany > Pruebas4DGeoservices > TEST-RAKAN > submeshes

Nombre	Fecha de modifica...	Tipo	Tamaño
textures	09/03/2018 11:07	Carpeta de archivos	
Inmaculada 450.cfg	09/10/2017 10:14	Archivo CFG	1 KB
Inmaculada 450.dat	09/10/2017 10:14	Archivo DAT	7.742 KB

Ilustración 92 Carpeta donde se encuentran los modelos secundarios de un proyecto

- b) Entrar en la carpeta textures del proyecto TEST-RAKAN y eliminar del nombre de los archivos "Inmaculada 450-base.ctx" e "Inmaculada 450-base.jpg" el sufijo "**-base**".
- c) Comprobamos el resultado:



Ilustración 93 Modelo base y modelo secundario de ejemplo

16. Referencias

- ❖ 4D Geoservices. (01 de 06 de 2017). *Inicio*. Recuperado el 10 de 05 de 2018, de <http://www.4dgeoservices.com/>
- ❖ 4D Geoservices. (01 de 06 de 2017). *Modelado 3D*. Recuperado el 10 de 05 de 2018, de <http://www.4dgeoservices.com/secModel3D.php>
- ❖ Cardozzo, D. R., & Academy, I. T. (2014). *Desarrollo de Software*. EISENBRAUNS.
- ❖ Díaz, A. D. (2017). *Unity 2017.x : curso práctico*. Ra-Ma S.A. Editorial y Publicaciones.
- ❖ Gagalowicz, A., & Philips, W. (2011). *Computer Vision/Computer Graphics Collaboration Techniques: 5th International Conference, MIRAGE 2011, Rocquencourt, France, October 10-11, 2011. Proceedings*. Springer Berlin Heidelberg.
- ❖ Hess, R. (2011). *Blender*. Anaya Multimedia.
- ❖ Jorquera Ortega, A. (2017). *Fabricación digital: Introducción al modelado e impresión 3D*. Ministerio de Educación de España.
- ❖ Korpela, J. (2006). *Unicode Explained*. O'Reilly Media.
- ❖ Kurman, M., & Lipson, H. (2014). *La revolución de la impresión 3D*. Anaya Multimedia.
- ❖ Lahoz, J. G., Calderón, A. B., & Aguilera, D. G. (2010). *Reconstrucción 3D y realidad virtual en criminología*. Ediciones Universidad de Salamanca.
- ❖ Martel, A. (2014). *Gestión Práctica de Proyectos Con Scrum: Desarrollo de Software Ágil Para El Scrum Master*. CreateSpace Independent Publishing Platform.
- ❖ Ogre3D Team. (27 de 4 de 2016). *Ogre3D website*. Recuperado el 14 de 05 de 2018, de Ogre3D website: <https://www.ogre3d.org/>
- ❖ Schenk, T., & Catalunya, I. C. (2002). *Fotogrametría digital*. Marcombo.
- ❖ Subero, I. G. (2014). *Libro de Unity 3D en castellano Volumen 1: La interfaz de Unity 3D al completo*. TRINIT Asoc. Informáticos de Zaragoza.

