	UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA			
	Documento	Código	Fecha	Revisión
	FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO	F-AC-DBL-007	10-04-2012	A
Dependencia	Aprobado		Pág.	
DIVISIÓN DE BIBLIOTECA	SUBDIRECTOR ACADÉMICO		i(147)	

RESUMEN – TRABAJO DE GRADO

AUTORES	HAROLD STIBENZON TORRADO PEREZ JESÚS ALFONSO BALLESTEROS MADARIAGA
FACULTAD	INGENIERÍAS
PLAN DE ESTUDIOS	INGENIERÍA DE SISTEMAS
DIRECTOR	BADWIN ARÉVALO VERA
TÍTULO DE LA TESIS	DISEÑO E IMPLEMENTACIÓN DE UN RECORRIDO VIRTUAL ACCESIBLE A TRAVÉS DE UN NAVEGADOR WEB QUE FACILITE EL RECONOCIMIENTO DEL CAMPUS DE LA UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA

RESUMEN

(70 palabras aproximadamente)

EL DISEÑO E IMPLEMENTACIÓN DE UN RECORRIDO VIRTUAL ACCESIBLE A TRAVÉS DE UN NAVEGADOR WEB QUE FACILITE EL RECONOCIMIENTO DEL CAMPUS DE LA UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA ES EL OBJETIVO PRINCIPAL DE ESTE DOCUMENTO Y EN EL SE PLASMAN TODAS LAS ETAPAS QUE SE LLEVARON A CABO PARA LA REALIZACIÓN DE ESTE, ASÍ COMO LAS HERRAMIENTAS UTILIZADAS DURANTE EL DESARROLLO DE CADA UNA DE ELLAS.

CARACTERÍSTICAS

PÁGINAS: 137	PLANOS: 0	ILUSTRACIONES: 0	CD-ROM: 1
---------------------	------------------	-------------------------	------------------



DISEÑO E IMPLEMENTACIÓN DE UN RECORRIDO VIRTUAL ACCESIBLE A TRAVÉS
DE UN NAVEGADOR WEB QUE FACILITE EL RECONOCIMIENTO DEL CAMPUS DE
LA UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA

AUTORES:

HAROLD STIBENZON TORRADO PÉREZ

JESÚS ALFONSO BALLESTEROS MADARIAGA

Trabajo de Grado para Optar el Título de Ingeniero de Sistemas

Director

BADWIN ARÉVALO VERA

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SISTEMAS

Ocaña, Colombia

Abril de 2016

Índice

Capítulo 1: Diseño e Implementación de un Recorrido Virtual Accesible a través de un Navegador Web que Facilite el Reconocimiento del Campus de la Universidad Francisco De Paula Santander Ocaña	1
1.1 Planteamiento del Problema	1
1.2 Formulación del Problema	2
1.3 Justificación.....	2
1.4 Objetivos.....	3
1.4.1 General.	3
1.4.2 Específicos	4
1.5 Delimitación	4
1.5.1 Geográfica.....	4
1.5.2 Temporal.	5
1.5.3 Conceptual.	5
1.5.4 Operativa.....	5
 Capítulo 2: Marco Referencial	 7
2.1 Marco Histórico.....	7
2.2 Marco Contextual	10
2.2.1 Reseña Histórica.	10
2.2.2 Misión.	13
2.2.3 Visión.	14
2.2.4 Objetivos	14
2.3 Marco Conceptual	15
2.3.1 Modelado 3D.....	15
2.3.2 Clasificación de los Sistemas de Modelado.....	16
2.3.3 Computación Gráfica.	18
2.3.4 Generalidades sobre Optimizaciones de Modelos 3D.	20
2.3.5 Recorridos Virtuales.	23
2.3.6 Tecnologías disponibles.....	24
2.3.7 Herramientas de software empleadas en este proyecto.....	27
2.4 Marco Legal.....	29
2.4.1 Artículo 71 de la Constitución Política Colombiana.	29
2.4.2 Ley 842 de 2003.....	30
2.4.3 Licencias para Software Libre.	30
2.4.4 Digital Millennium Copyright Act (DMCA).	31
 Capítulo 3: Diseño Metodológico.....	 32
3.1 Tipo de Investigación	32
3.2 Población	32
3.3 Muestra	33
3.4 Técnicas e Instrumentos de Recolección de la Información	34

3.5 Análisis de la Información.....	34
3.6 Metodología de Desarrollo.....	52
Capítulo 4: Presentación de resultados.....	55
4.1 Modelar estructuras y objetos de los edificios “La Casona” y “Anexos”, así como el personaje guía del recorrido, con el fin de tener los prototipos en 3D.....	55
4.1.1 Recopilación de planos y material fotográfico.	55
4.1.2 Modelado en Blender.....	55
4.1.3 Texturizado.	63
4.2 Organizar el entorno del recorrido virtual con el fin de integrar y dar interactividad a los objetos creados.....	64
4.2.1 Creación de terreno.	65
4.2.2 Importación de los modelos 3D a Unity.	81
4.2.3 Creación del Mini-Mapa.....	83
4.2.4 Optimizaciones.....	85
4.2.5 Creación de la escena del dialogo de Santanderito.	90
4.2.6 Creación de la Escena de Inicio y la Escena del Menú de Inicio.....	93
4.3 Incorporar la Aplicación Desarrollada del Recorrido Virtual en un Sitio Web, Facilitando el Acceso por Medio de Internet.	99
4.3.1 Parámetros de Compilación.	99
4.3.2 Incorporación de la Aplicación del Recorrido Virtual en el Sitio Web.	102
Capítulo 5: Conclusiones.....	104
Capítulo 6: Recomendaciones.....	106
Referencias.....	108
Apéndices.....	111

Lista de Figuras

Figura 1. Mapeado cilíndrico.	25
Figura 2. Mapeado esférico	25
Figura 3. Escena en formato VRML	26
Figura 4. Nivel de conocimiento que el estudiante tiene sobre la ubicación de la Universidad.....	36
Figura 5. Nivel de dificultad que el estudiante ha tenido para ubicar oficinas en la Casona o Anexos.....	37
Figura 6. Nivel de conocimiento que el estudiante tiene sobre las herramientas que la Universidad ofrece para facilitar la ubicación de las dependencias.	39
Figura 7. Calificación que el estudiante le da a las herramientas a través de las cuales la universidad actualmente ofrece información de las dependencias.....	41
Figura 8. Frecuencia de utilización de los mapas de ubicación por parte del estudiante....	42
Figura 9. Calificación, en cuanto a practicidad y facilidad de uso, de los mapas de ubicación de la Universidad.	44
Figura 10. Nivel de satisfacción por parte del estudiante frente a la inducción ofrecida por la Universidad en cuanto a organización del campus y sus dependencias.	45
Figura 11. Nivel de satisfacción del estudiante respecto a la inducción ofrecida por la Universidad.....	47
Figura 12. Aceptación del estudiante frente a una herramienta accesible vía web que le permita conocer y recorrer los edificios Casona y Anexos de la UFPSO.	48
Figura 13 Nivel de interés del estudiante por dar a conocer la Universidad.	50

Figura 14. Experiencias por parte del estudiante en cuanto a visitas virtuales a instituciones.....	51
Figura 15. Importar archivo DXF.....	56
Figura 16. Importar Autodesk DXF	57
Figura 17. Plano importado.....	57
Figura 18. Mejoras a la malla.....	58
Figura 19. Vista de alambres, edificio Casona	59
Figura 20. Vista de alambres, edificio Anexos Académicos.....	59
Figura 21. Vista de alambres, caminos y edificios de clases de la Universidad Francisco de Paula Santander Ocaña.....	60
Figura 22. Modelado de objetos y divisiones.....	61
Figura 23. Vistas del personaje Santanderito	61
Figura 24. Modelo 3d del personaje Santanderito	62
Figura 25. Vista de esqueleto.	63
Figura 26. Grupos de vértices y formas clave.....	63
Figura 27. Textura Santanderito.....	64
Figura 28. Silla con texturas.....	64
Figura 29. Vista superior en Blender del Plano general simplificado de la Universidad....	66
Figura 30. Selección del terreno alrededor de la UFPSO en el programa SketchUp.....	67
Figura 31. Plano generado.....	67
Figura 32. Geografía del terreno alrededor de la UFPSO.....	68
Figura 33. Maya del terreno importada en Blender.....	69
Figura 34. Representación por cubos de algunas zonas de la Universidad.	70

Figura 35. Alineación de los dos modelos.	71
Figura 36. Importación del asset del terreno en Unity.	72
Figura 37. Terrenos dentro de Unity, izquierda Maya Inicial del terreno y derecha GameObject generado con el script.....	73
Figura 38. Alineación de los terrenos dentro de Unity.....	74
Figura 39. Integración del GameObject Terrain y el asset del plano de la Universidad. ..	75
Figura 40. Herramientas para trabajar con terrenos ofrecidas por Unity en el Panel del Inspector.	76
Figura 41. Terreno obtenido luego de aplicar texturas y ubicar árboles.	77
Figura 42. Imágenes 360 grados para skybox.	78
Figura 43. Cortes realizados al terreno inicial.....	79
Figura 44. Terreno final.....	80
Figura 45. Vista superior del terreno con el skybox.....	81
Figura 46. Vista desde la superficie del terreno con el skybox.	81
Figura 47. Vista virtual de una oficina	82
Figura 48. Vista exterior del campus.....	82
Figura 49. Vista Superior del Terreno.....	83
Figura 50. Vista normal del Mini-mapa	84
Figura 51. Mini-mapa maximizado al presionar la tecla P	85
Figura 52. Ubicación de los triggers en el edificio de Anexos Académicos.....	86
Figura 53. Script desactivado. 3.2 M de vértices en la escena.	87
Figura 54. Estadísticas de rendimiento de la escena con el script desactivado.....	87
Figura 55. Script activado. 690 K vértices en la escena.	88

Figura 56. Estadísticas de rendimiento de la escena con el script activado.	88
Figura 57. Parámetros de Configuración de Iluminación en Unity.	89
Figura 58. Vista de una oficina con iluminación.....	90
Figura 59. Sincronización de fonemas con Papagayo	91
Figura 60. Animación en Unity	92
Figura 61. Escena del dialogo introductorio.....	93
Figura 62. Escena del recorrido del Campus.....	94
Figura 63. Escena dialogo de Santanderito en el auditorio Fabio Amaya.	94
Figura 64. Escena de Inicio de la aplicación.	95
Figura 65. Escena del menú de inicio.....	96
Figura 66. Panel de ayuda del menú de inicio.....	97
Figura 67. Panel de configuración del menú de inicio.	98
Figura 68. Panel de créditos del menú de inicio.....	98
Figura 69. Fase de carga del WebPlayer.	100
Figura 70. Fase de carga del WebGL.	100
Figura 71. Recorrido virtual hospedado en el host campusvirtual.ufpso.edu.co.	103

Lista de Tablas

Tabla 1. Descripción de variables.	33
Tabla 2. Nivel de conocimiento que el estudiante tiene sobre la ubicación de la Universidad.....	35
Tabla 3. Nivel de dificultad que el estudiante ha tenido para ubicar oficinas en la Casona o Anexos.....	37
Tabla 4. Nivel de conocimiento que el estudiante tiene sobre las herramientas que la Universidad ofrece para facilitar la ubicación de las dependencias.....	39
Tabla 5. Calificación que el estudiante le da a las herramientas a través de las cuales la universidad actualmente ofrece información de las dependencias.....	40
Tabla 6. Frecuencia de utilización de los mapas de ubicación por parte del estudiante.	42
Tabla 7. Calificación por parte del estudiante, en cuanto a practicidad y facilidad de uso, de los mapas de ubicación de la Universidad.....	43
Tabla 8. Nivel de satisfacción por parte del estudiante frente a la inducción ofrecida por la universidad en cuanto a organización del campus y sus dependencias.....	45
Tabla 9. Nivel de satisfacción por parte del estudiante frente a la inducción ofrecida por la Universidad.....	46
Tabla 10. Aceptación del estudiante frente a una herramienta accesible vía web que le permita conocer y recorrer los edificios Casona y Anexos de la UFPSO.	48
Tabla 11. Nivel de interés del estudiante por dar a conocer la Universidad.	49
Tabla 12. Experiencias por parte del estudiante en cuanto a visitas virtuales a instituciones.....	51
Tabla 13. Configuración de calidad con respecto al rendimiento del computador.	102

Lista de Apéndices

Apéndice A. Plano General Universidad Francisco de Paula Santander Ocaña.	112
Apéndice B. Plano por bloques de la Universidad Francisco de Paula Santander Ocaña.	113
Apéndice C. Script Object2Terrain	114
Apéndice D. Script UnirTerreno	118
Apéndice E. Script AdminAssetBundle	119
Apéndice F. Scripts para controlar la visibilidad de los objetos dentro del recorrido.....	121
Apéndice G. Libreto del dialogo de Santanderito	123
Apéndice H. Scripts para generar animaciones de la sincronización labial de Santanderito.....	126
Apéndice I. Script Mini-Mapa.....	131

Diseño e Implementación de un Recorrido Virtual Accesible a través de un Navegador Web que Facilite el Reconocimiento del Campus de la Universidad Francisco De Paula Santander Ocaña

1.1 Planteamiento del Problema

La universidad Francisco de Paula Santander Ocaña, específicamente en la última década y debido a sus políticas de mejoramiento de la calidad y de los procesos, ha venido sufriendo cambios en pro de hacerla mejor y competitiva en un medio cada vez más exigente.

Lo anterior ha permitido una ampliación tanto de la planta física así como de la cobertura de la población estudiantil, pero a su vez ha conducido a que internamente se presenten casos en los cuales estudiantes, visitantes e incluso profesores, pueden tener problemas para precisar la ubicación de un sitio o dependencia, pues la localización de éstas está sujeta a constantes cambios a causa de ampliaciones y mejoras en la infraestructura.

En la actualidad existen mapas de ubicación dispuestos en varios sectores del Campus Universitario, los cuales facilitan información sobre la localización de las diferentes zonas; dicha herramienta es accesible de forma presencial.

Día a día, la Universidad continúa su proceso de mejoramiento, aumentará la población estudiantil situación que conlleva a plantear alternativas de solución a un problema creciente, y contrarrestar el rezago frente a universidades que en circunstancias similares ya han implementado soluciones.

1.2 Formulación del Problema

¿El diseño e implementación de un recorrido virtual accesible a través de un navegador Web facilitará el reconocimiento del campus de la Universidad Francisco de Paula Santander Ocaña?

1.3 Justificación

Tomando como base una encuesta realizada a una muestra de estudiantes de los semestres primero, segundo y tercero, del primer semestre de 2014 por los investigadores del presente proyecto, y que dio como resultado la dificultad de ubicar las zonas y dependencias con las cuales cuenta la universidad; surge entonces la necesidad de un instrumento de apoyo de fácil acceso y manipulación a través del cual cualquier persona interesada pueda observar y recorrer la infraestructura física de los edificios Casona y Anexos de la Universidad sin necesidad de estar presente físicamente, es decir, permitiendo el acceso remoto a éste.

Una opción, desde la ingeniería de sistemas, es el desarrollo de un entorno virtual cuyo escenario sea el campus de la universidad Francisco de Paula Santander Ocaña y por medio de un recorrido virtual se pueda mostrar la planta física de los edificios anteriormente mencionados, de una forma agradable e interactiva.

Las razones para optar por el desarrollo de un recorrido virtual de la Universidad Francisco de Paula Santander Ocaña son varias, entre estas se encuentra la necesidad de un sistema que permita dar a conocer nuestra Alma Máter a estudiantes, profesores, visitantes y público en general que deseen estar al tanto de los cambios en infraestructura física y ubicación de dependencias realizados hasta la fecha; ofreciendo así una gran ayuda al momento de realizar diligencias, asistir a clases o conocer la localización de los diversos lugares.

De la misma manera, la generación de un incentivo por la carrera y la universidad a estudiantes que se encuentran a la expectativa de una motivación para iniciar o continuar su formación en la UFPSO.

1.4 Objetivos

1.4.1 General. Diseñar e implementar un recorrido virtual accesible a través de un navegador Web que facilite el reconocimiento del campus de la Universidad Francisco de Paula Santander Ocaña, sede El Algodonal.

1.4.2 Específicos

- Modelar estructuras y objetos de los edificios “La Casona” y “Anexos”, así como el personaje guía del recorrido, con el fin de tener los prototipos en tres dimensiones (3D a partir de ahora), los cuales se realizarán con la o las herramientas seleccionadas.
- Organizar el entorno del recorrido virtual con el fin de integrar y dar interactividad a los objetos creados, mediante la utilización de las herramientas que se seleccionen para este fin.
- Incorporar la aplicación desarrollada del recorrido virtual en un sitio web, facilitando el acceso por medio de internet.

1.5 Delimitación

1.5.1 Geográfica. El Diseño e implementación de un recorrido virtual accesible a través de un navegador Web que facilite el reconocimiento del campus de la Universidad Francisco de Paula Santander Ocaña, se llevará a cabo en la Universidad Francisco de Paula Santander Ocaña, vía Acolsure sede EL Algodonal.

1.5.2 Temporal. El Diseño e implementación de un recorrido virtual accesible a través de un navegador Web que facilite el reconocimiento del campus de la Universidad Francisco de Paula Santander Ocaña, se llevará a cabo en un periodo de tres (4) meses a partir de la aprobación del proyecto.

1.5.3 Conceptual. En el marco conceptual se desarrollarán varios tópicos relacionados a los entornos 3D: En un primer aspecto se analizarían los conceptos y temas relacionados con los recorridos virtuales y realidad virtual. Un segundo aspecto abordará técnicas y prácticas para optimizar los modelos 3D, herramientas que se emplearán en cada etapa de desarrollo. Finalmente se hará referencia a información básica acerca de la Universidad Francisco de Paula Santander Ocaña.

1.5.4 Operativa. Para llevar a cabo los procesos de modelado y texturizado de los diferentes edificios de la universidad Francisco de Paula Santander Ocaña se empleará el programa Blender como herramienta de desarrollo.

Para la creación del entorno y animación del recorrido virtual se empleara el programa Unity. Esta herramienta es licenciada, razón por la cual se aclara que si la universidad no adquiere la licencia para Unity Pro, se entregará una versión final del proyecto con una licencia libre, que está sujeta a limitaciones.

Como primera etapa del proyecto, “Diseño e implementación de un recorrido virtual accesible a través de un navegador Web que facilite el reconocimiento del campus de la Universidad Francisco de Paula Santander Ocaña”, se propone la creación de modelos 3D de los siguientes bloques de la universidad.

Bloques que harán parte del grupo de modelos son:

- Bloque 01 “La Casona”.
- Bloque 19 “Anexos”.

Los números de los bloques empleados para referirse a los edificios anteriormente mencionados se obtuvieron de los planos suministrados por la oficina de Planeación de la universidad.

Las mallas de la planta física de la universidad Francisco de Paula Santander Ocaña producto del proceso de modelado 3D, permitirán un alto grado de interacción, que va desde una vista exterior hasta la posibilidad de explorar los espacios interiores del modelo, como por ejemplo, abrir puertas, entre otros.

Este proyecto cubre lo que se propone en el inciso anterior y constituiría la primera etapa; cualquiera de las ampliaciones propuestas deberán ser desarrolladas por el o los estudiantes que estén interesados en continuar con la ampliación y desarrollo de este proyecto.

Marco Referencial

1.6 Marco Histórico

Para la revisión de experiencias y los proyectos realizados que tuvieran relación con el presente proyecto, se procedió a buscar en la Web a nivel local, nacional e internacional.

A nivel local se indagó en la Universidad Francisco de Paula Santander Ocaña y se encontró que no se han realizado experiencias, razón por la cual no existe un referente.

En el país, se reconoce el trabajo: Análisis, Diseño e Implementación de un Software para Visitas Virtuales 3D Interactivas por instalaciones de la Universidad Tecnológica de Pereira, desarrollado por Mauricio Alejandro Valencia Martínez y David Vargas Valencia, en Pereira, en el año 2007. Para el desarrollo del proyecto se partió del plano general de La Universidad, el cual luego de un proceso de simplificación fue importado a Google SketchUp, programa mediante el cual se crearon los modelos tridimensionales de los edificios y se llevó a cabo la texturización de estos; a continuación las mayas terminadas, junto con su correspondiente material y textura, se exportaron en un formato aceptado por el motor gráfico Ogre. Al final de este proyecto se llega a las siguientes conclusiones: el desarrollo de aplicaciones gráficas que usan tecnología 3D requiere especificaciones de hardware que son determinadas por las herramientas empleadas para su creación. La creación y texturización de modelos tridimensionales que simulan objetos de la realidad es un proceso complejo y esto hace necesario evaluar las herramientas existentes para

así elegir las adecuadas para el tipo de proyecto y el tipo de objeto 3D que se desee modelar, la aplicación debe contar con una documentación clara que le permita al usuario aprovechar al máximo toda la funcionalidad de la misma. (Valencia Martínez & Vargas Valencia, 2007). Del proyecto anteriormente mencionado se tendrá en cuenta el proceso de simplificación mediante el cual se obtienen los planos que servirán como base para el desarrollo de los modelos 3D.

En la búsqueda a nivel nacional, se encontró también el proyecto titulado Visita virtual, documentación e información de la Escuela Técnica Superior de Ingeniería de Telecomunicación, realizado por Víctor Ros Escribano en Cartagena, en el año 2008, proyecto que tuvo como objetivo desarrollar una aplicación que diera a conocer la Universidad Politécnica de Cartagena, concretamente la Escuela Técnica Superior de Ingeniería de Telecomunicación, con sede en el Cuartel de Antigones, haciendo uso de técnicas de modelado de Realidad Virtual (RV). En el informe se analiza que las técnicas de RV ofrecen un entorno en el cual el usuario tiene grandes posibilidades de interacción y que la aparición del Lenguaje de Modelado de Realidad Virtual (VRLM), Java 3D, Blender y de plugin para estos lenguajes, ha hecho posible la construcción de mundos virtuales incluso accesibles a través de Internet. Las fases que se siguieron en la elaboración fueron: estudio de los diversos software a utilizar, análisis y diseño de la aplicación, implementación lógica de la visita virtual. (Ros Escribano, 2008). De este proyecto se tomará como modelo a seguir las fases empleadas en proceso de desarrollo de la aplicación.

A nivel internacional por los resultados obtenidos en la Web, se encontró que Reynolds León Guerra habla acerca de la Aplicación de la realidad virtual no inmersiva para Ingenieros Agrícolas, trabajo que aborda la utilización de la realidad virtual no inmersiva en el campo de la ingeniería, muy utilizado hoy en día por diferentes universidades y centros de investigación para el desarrollo de sus investigaciones y aplicación en la docencia. El trabajo se centra en el modo de uso de la realidad virtual por especialistas no informáticos que no tengan conocimientos de programación, donde se desarrolla una metodología que permite crear mundos virtuales sin la utilización de técnicas de programación y de fácil desarrollo ya que utilizan herramientas muy conocidas en el campo de los ingenieros como son los sistemas CAD(Diseño Asistido por Computador)/CAE (Ingeniería Asistida por Computadora); aunque no se deja de mencionar aspectos importantes sobre Web y VRML, que son necesarios para una mejor comprensión de la metodología elaborada y consideraciones especiales que debe tenerse en cuenta para crear mundos virtuales o escenas virtuales. (Leon Guerra, 2012). Del artículo citado anteriormente se aprovecharan la experiencias en la utilización de herramientas CAD y CAE, y de igual manera se tendrán en cuenta las metodologías y consideraciones plasmadas por el autor.

Finalmente a nivel internacional se encontró el trabajo de Ramón Brun Amaya: Generación y visualización inmersiva del estadio olímpico de Madrid, para el desarrollo del modelado del Estadio Olímpico de la Comunidad de Madrid en el cual se partió de un plano en planta de todo el recinto, la generación del modelo tridimensional, se ha afrontado subdividiendo el estadio en diversas partes que se han generado por separado y que posteriormente han sido integradas en un conjunto. Esta manera de abordar el problema ha servido para simplificar la complejidad de cada parte a modelar, pero debido a las formas complejas del conjunto, ha requerido posteriormente

de una etapa de unión y adaptación de las diversas partes para lograr un encaje perfecto de las mismas. Como resultado de la metodología aplicada en el desarrollo de la aplicación se presentaron dificultades a la hora de integrar las diferentes partes que componían el estadio, también debido al empleo de diferentes herramientas en las etapas de desarrollo del proyecto se presentaron algunos inconvenientes con la exportación del modelo de una a otra. (Brun Amaya, 2010). Este trabajo nos aporta un modelo de trabajo en el cual se puede lograr un modelado de estructuras complejas subdividiendo el modelo inicial en modelos con menor complejidad, lo que facilitará la consecución del objetivo final.

1.7 Marco Contextual

El desarrollo del trabajo investigativo se llevó a cabo en el campus de la universidad Francisco de Paula Santander seccional Ocaña, para la cual daremos a conocer un poco de su reseña historia, misión, visión y objetivos de la entidad, para una mejor comprensión del lector.

1.7.1 Reseña Histórica. En noviembre de 1973 se suscribió un contrato para la realización de un estudio de factibilidad denominado "un centro de educación superior para Ocaña" que fue terminado y sugirió la creación pronta de un programa de educación a nivel de tecnología en énfasis en ciencias sociales, matemáticas y física. En diciembre de ese mismo año, el rector de la Universidad Francisco de Paula Santander, José Luís Acero Jordán, le envió copia de dicho estudio al ICFES, Instituto que conceptúo que el proyecto para abrir el centro de estudios en Ocaña, era recomendable.

Según Acuerdo No. 031 del 18 de Julio de 1974, por parte del Consejo Superior de la Universidad Francisco de Paula Santander Cúcuta, se crea la Universidad Francisco de Paula Santander Ocaña, como máxima expresión cultural y patrimonio de la región; como una entidad de carácter oficial seccional, con AUTONOMÍA administrativa y patrimonio independiente, adscrito al Ministerio de Educación Nacional.

Su primer coordinador el doctor Aurelio Carvajalino Cabrales, buscó un lugar adecuado para funcionar la sede, en los claustros Franciscanos al costado del templo de la Gran Convención y con las directivas del colegio José Eusebio Caro, se acordó el uso compartido del laboratorio de física.

En 1975 comenzó la actividad académica en la entonces seccional de la Universidad Francisco de Paula Santander con un total de 105 estudiantes de Tecnología en Matemáticas y Física, y su primera promoción de licenciados en Matemáticas y Física se logró el 15 de diciembre de 1980.

La consecución de 27 hectáreas de la Hacienda El Rhin, en las riberas del Río Algodonal, en comodato a la Universidad por 50 años, que la antigua Escuela de Agricultura de Ocaña cedió a la Universidad, permitió la creación del programa de Tecnología en Producción Agropecuaria, aprobado por el Consejo Superior mediante el Acuerdo No. 024 del 21 de agosto de 1980, y luego el ICFES otorgó la licencia de funcionamiento el 17 de febrero del año siguiente. Luego se crean las Facultades.

La Facultad de Ciencias Agrarias y del Ambiente, fue creada según Acuerdo 084 del 11 de septiembre de 1995 conformada por los departamentos de Ciencias Agrícolas y del Ambiente y el departamento Ciencias Pecuarias junto a los programas académicos de Tecnología Agropecuaria (Acuerdo N° 024 del 21 de agosto de 1980), Zootecnia (Acuerdo N° 057 y 058 del 27 de junio de 2007), e Ingeniería Ambiental (Acuerdo 089 del 9 de octubre 1995).

La Facultad de Ciencias Administrativas y Económicas, fue creada según Acuerdo No. 008 del 05 de marzo de 2003; está conformada por el departamento de Ciencias Administrativas y Departamento de ciencias Contables y Financieras. Están adscritos los programas académicos de Tecnología en Administración Comercial y Financiera (Acuerdo No, 024 del 29 de Junio de 1988 y con la Resolución 5243 del 05 de Septiembre del 2006 del MEN), Administración de Empresas (Acuerdo No, 024 del 29 de Junio de 1988) y la profesionalización (Acuerdo No. 118 del 16 de Noviembre de 1994); Contaduría Pública (Acuerdo No. 007 del 05 de Marzo de 2003 y según resolución 3388 del 23 de Diciembre del 2003 del MEN). Así mismo, según Acuerdo No. 0087 del 15 de diciembre del 2005 se aprueba por Ciclos Propedéuticos el Plan de Estudio de la Técnica Profesional en Administración Comercial Y Financiera, según Resolución 101 del 18 de enero de 2007 del MEN.

La Facultad de Ingenierías fue creada según acuerdo 007 del 20 de febrero de 2006, conformada con los departamentos de Ingeniería Civil, Ingeniería Mecánica y el departamento de Sistemas e Informática. Con los registros calificados de los programas completos de acuerdo a la Resolución 2909 de julio 21 de 2005 para el programa de Ingeniería Civil e Ingeniería

Mecánica (Resolución 2908 de julio 21 de 2005), Ingeniería de Sistemas (Resolución 7062 de noviembre 10 de 2006). La creación de los Técnicos Profesionales en Telecomunicaciones con registro calificado (Resolución 5366 de agosto 25 de 2008) y el Técnico profesional en Informática con registro calificado (Resolución 4613 de julio 18 de 2008).

La Facultad de Educación, Artes y Humanidades de la Universidad Francisco de Paula Santander Ocaña fue creada según acuerdo 063 del 20 de noviembre de 2006, está conformada con los departamentos: de Matemáticas, física y computación y el departamento de Humanidades. Según el Acuerdo No. 010, marzo 29 de 2004 se crea el plan de estudios del programa de Comunicación Social, Derecho con registro calificado (Resolución 10185 de noviembre 22 de 2010). En el mes de noviembre de 2005, se suscribió el convenio de asociación No. 1744/05 con el Ministerio de Cultura, con el objeto de apoyar el proceso de estructuración académica de la Escuela de Bellas Artes. (Universidad Francisco de Paula Santander Ocaña, s.f.)

1.7.2 Misión. La Universidad Francisco de Paula Santander Ocaña, institución pública de educación superior, es una comunidad de aprendizaje y autoevaluación en mejoramiento continuo, comprometida con la formación de profesionales idóneos en las áreas del conocimiento, a través de estrategias pedagógicas innovadoras y el uso de las tecnologías; contribuyendo al desarrollo nacional e internacional con pertinencia y responsabilidad social. (Universidad Francisco de Paula Santander Ocaña, s.f.)

1.7.3 Visión. La Universidad Francisco de Paula Santander Ocaña para el 2019, será reconocida por su excelencia académica, cobertura y calidad, a través de la investigación como eje transversal de la formación y el uso permanente de plataformas de aprendizaje; soportada mediante su capacidad de gestión, la sostenibilidad institucional, el bienestar de su comunidad académica, el desarrollo físico y tecnológico, la innovación y la generación de conocimiento, bajo un marco de responsabilidad social y ambiental hacia la proyección nacional e internacional. (Universidad Francisco de Paula Santander Ocaña, s.f.)

1.7.4 Objetivos

Desarrollo de Talento Humano

La Universidad mantendrá su preocupación por el desarrollo del talento humano (Estudiantes, Docentes y Administrativos) para que se integren con entusiasmo a los desafíos de la organización y el entorno en general.

Modernización Tecnológica

En los próximos 3 años, deberá concluir la modernización de todos los medios de operación para garantizar la productividad y el permanente control del proceso, con máxima flexibilidad y calidad académica y administrativa.

Fortalecimiento Investigación y Extensión.

La universidad considera de vital importancia el liderazgo en el desarrollo tecnológico, para ello propone 2 objetivos fundamentales; la Revitalización de la Investigación y la búsqueda de nuevas tecnologías para el desarrollo de los sectores social y productivo.

Crecimiento de Nuevas Líneas de Productos.

Especialmente en el desarrollo de postgrados y Planes de Estudio, Educación Continuada y Universidad a Distancia. (Universidad Francisco de Paula Santander Ocaña, s.f.)

1.8 Marco Conceptual

Para efectos del presente proyecto, y después de una amplia búsqueda, no se encontraron teorías que respalden su desarrollo, por lo tanto, el presente proyecto fue fundamentado en los conceptos citados a continuación.

1.8.1 Modelado 3D. En informática gráfica, un modelo es un conjunto de datos estructurado, en general con un importante componente matemático, que utiliza una aplicación informática para describir los objetos. Los modelos hacen una descripción de aspectos como la

geometría del objeto, es decir forma geométrica y posición espacial, la topología, correspondiente al detalle de las partes que lo componen como número de caras y las propiedades de aspecto, las cuales se refieren al comportamiento del modelo frente a la iluminación, la elasticidad entre otras. (Hernández Abab, Ochoa Vives, Hernández Abab, & Farré Desongles, 2004, pág. 5)

1.8.2 Clasificación de los Sistemas de Modelado. Existen varios tipos de sistemas de modelado y la elección de cual usar para un determinado proyecto dependerá de lo que se quiera lograr, es así como se enumeran principalmente cuatro tipos de sistemas de modelado, de alambre, superficies, volúmenes y especiales.

El modelado de alambre es uno de los primeros sistemas de modelado diseñados durante los años setenta, en donde la representación del objeto se hace a través de puntos conectados entre sí mediante líneas rectas y curvas, por ser un modelo sencillo no necesita de gran cantidad de recursos del sistema y resulta ser fácil de utilizar para construir modelos geométricos con él. (Hernandez Abab et al., 2004, pág. 8)

Cuando solo se representa la superficie exterior de los objetos se habla de modelos de superficies, este modelo no aporta información acerca del volumen y los datos para su generación pueden ser obtenidos por diferentes métodos, tales como modelado manual, escáner tridimensional o procedimientos de pre-cálculo. (Hernandez Abab et al., 2004, pág. 9)

El grupo correspondiente a modelo de volúmenes está constituido por aquellos sistemas de modelado en los que la representación generada es en función del volumen del objeto que se desea representar.

Por último se encuentran los modelos especiales que engloban sistemas de modelado como fractales, partículas, esferas flexibles. Los primeros se basan en la aplicación repetitiva de funciones de transformación a puntos de una región del espacio, este tipo de modelado es especialmente útil para generar superficies indeterminadas como lo son los terrenos, montañas entre otras. Una partícula es un elemento que tiene ciertas características tales como una forma, tiempo de vida, trayectoria y que se puede ver influida por fuerzas exteriores como la gravedad, esas características hacen que sean adecuadas para modelar fenómenos naturales que cambian en el tiempo, el fuego, humo entre otros son ejemplos en los cuales este modelo resulta el ideal, en esta misma categoría se encuentran las esferas flexibles, el proceso de modelado consiste en la fusión de esferas a las cuales se les puede cambiar la posición, diámetro y fuerza de interacción, es así como se puede controlar la forma en que interactúan entre sí, este tipo de modelado es adecuado para cuando se quieren representar fluidos. (Hernandez Abab et al., 2004, pág. 21)

Es importante resaltar el uso específico de cada uno de estos sistemas de modelado, ya que, un uso indebido de estos puede resultar en un consumo excesivo de recursos los cuales podrían ser mejor aprovechados utilizando las técnicas correctas según sea requerido. De lo mencionado anteriormente y para el caso del presente proyecto se seleccionó el modelo de alambre y el

modelo de superficies, ya que se basan en formas simples y de poco consumo, que permite lograr una experiencia fluida sin la necesidad de una gran cantidad de recursos computacionales.

1.8.3 Computación Gráfica. El desarrollo de gráficos por ordenador, es el campo de la informática visual donde se utilizan computadoras tanto para generar imágenes sintéticas como integrar o cambiar la información visual y espacial probada del mundo real. La computación gráfica comprende una gran variedad de técnicas que pueden ser agrupadas de acuerdo al número de dimensiones que se empleen en la representación del modelo geométrico a visualizar, la computación gráfica 2D y La computación gráfica 3D.

Cuando una imagen digital parte de modelos geométricos bidimensionales se habla de la computación gráfica 2D y son principalmente empleadas en interfaces gráficas de usuario y en aquellas aplicaciones que se desarrollan a partir de tecnologías de impresión y dibujo, como tipografía, cartografía, dibujo técnico, ... Las principales categorías de la computación gráfica 2D son los gráficos vectoriales y de rasterización. Los gráficos vectoriales emplean primitivas geométricas basadas en ecuaciones matemáticas, formadas por puntos, líneas, curvas y polígonos, con lo que representan las imágenes, por otro lado los gráficos de rasterización utilizan una matriz rectangular de píxeles para representar las imágenes, las cuales se pueden desplegar en cualquier dispositivo de salida.

La computación gráfica 3D trata acerca de la síntesis de una imagen bidimensional a partir de un modelo geométrico tridimensional. Las técnicas empleadas en la generación de las

imágenes teniendo en cuenta la complejidad de los cálculos empleados en esta son pre-rendering y real-time rendering. Las primeras son las utilizadas en la creación de animaciones de tipo foto-realista, mientras que las segundas se emplean en aplicaciones que requieren interactividad y para su utilización se requiere de un procesador de gráficos dedicado.

Cuando el trabajo se realiza utilizando las técnicas de rendering, debe tenerse en cuenta el modo en el que la luz se modele encontrándose diferentes categorías entre ellas la rasterización, Ray-tracing y Métodos de iluminación global.

La rasterización consiste en iterar en cada cuadro a través de todas las primitivas geométricas que conforman la escena para determinar, a partir del punto de vista del usuario cuales píxeles de la imagen se ven afectados. Cuando una aplicación demanda interactividad se utiliza el método de rasterización debido a que la mayoría de las GPU (Unidad de Procesamiento Gráfico) actuales utilizan principalmente este método convirtiéndolo así en el método más eficiente.

El método de Ray-tracing es una extensión del Ray-casting. En el método de Ray-casting el color de cada píxel de la imagen se calcula interceptando el color del primer objeto y un rayo imaginario que se proyecta desde el punto de vista, al píxel en cuestión; en el Ray-tracing se emplea esta técnica de modo recursivo, es decir, empleando rayos de refracción y reflexión a partir del punto en la superficie del objeto interceptado por el rayo inicial. Generalmente este cómputo se realiza tomando un número aleatorio de muestras (rayos de luz emanados de la

superficie de los objetos que intersecan el punto de vista) para luego promediarlas mediante técnicas de Monte Carlo.

Los Métodos de iluminación global, abarcan técnicas en las que se simula el modo en que las superficies iluminadas actúan a su vez como fuentes de iluminación de otras superficies, produciendo un efecto más realista en el que el ambiente de la escena parece ser mejor captado, esto se logra gracias al empleo de la teoría de elementos finitos.

Con lo tratado anteriormente se puede comprender la manera en la que se da forma a los objetos tridimensionales, a partir de cálculos matemáticos obteniendo así como resultado unas vistas de los diferentes objetos en 3D situados en la escena, representado en imágenes en 2D que pueden ser percibidas en una pantalla. A esto se le pueden agregar diferentes efectos con los cuales se lograría una panorámica más realista, como lo son: la iluminación, lo que permite darle a los objetos una mejor percepción de los detalles al agregarle sombras según el ángulo de iluminación. El campo de visión, es aquel que permite re presentar en el cuadro un efecto similar al campo de visión humano. Por otro lado el desenfoque nos daría la posibilidad de recalcar los objetos en los cuales estamos enfocados, haciendo que los que se encuentran a mayor o menor distancia tengan un aspecto borroso. Estos son algunos de los efectos que se manejan en este tipo de tecnologías.

1.8.4 Generalidades sobre Optimizaciones de Modelos 3D. En la informática gráfica se conoce como optimización al conjunto de procesos que se aplican a un programa para lograr

aprovechar al máximo los recursos de un sistema. Al desarrollar aplicaciones de gráficos por computadora, se encuentran principalmente tres categorías optimización, las Geométricas, las de Texturas y las de Cámara. (Franco Serrano, s.f)

Las primeras utilizan recursos como la sustitución de geometrías muy pesadas por geometrías ligera, dentro de esta categoría se encuentran los Niveles de detalle (LOD) que consiste en un conjunto de objetos que representan a un elemento con diferentes niveles de complejidad geométrica o de textura, propio de las aplicaciones tridimensionales para tiempo real, en segundo lugar encontramos la retopología, técnica que permite conseguir un modelo más simple partiendo de los vértices encontrados en un modelo que contiene gran cantidad de polígonos, con lo que se construye una malla de menor complejidad, para luego a través de procesos de pre-cálculo obtener mapas para simular detalles finos, por último en esta categoría se encuentra la Cajas de colisión, que consisten en una geometría invisible que rodea a los objetos de una escena, la cual tiene como fin obstruir el paso de la cámara y objetos que intenten atravesarlas.

Las Optimizaciones de Texturas consisten en la sustitución de geometrías de la escena empleando pre-cálculos de la iluminación, texturas aplicadas a polígonos planos y shaders. Dentro de este grupo de optimizaciones se encuentran, los Niveles de detalle en textura y MipMaps que parten de la idea de tener varias copias de una misma geometría, cada ejemplar de la geometría tiene una textura que conserva la calidad pero cambia de tamaño, estas geometrías se irán sustituyendo dependiendo de la distancia a la que se encuentre el punto de vista. También

se ubican los Shaders que eliminan polígonos innecesarios de la escena, pero que a su vez cargan el procesamiento gráfico si son excesivamente usados.

Texture Baking es el proceso que permite generar una textura con la información resultante del pre-cálculo de iluminación y materiales en texturas, mediante la utilización de las coordenadas de textura para determinar la posición de los objetos en el espacio UV. La textura que se crea con este proceso puede ser de difusión, de sombras, brillo, transparencia, ambient occlusion entre otros. Una geometría mapeada y texturizada con una textura producto del proceso mencionado presentara detalles como sombras, brillos o luces sin necesidad de que estos estén siendo calculados en tiempo real.

Como último elemento de las optimizaciones de texturas se encuentran los mapas auto orientables o billboards, estos son planos a los que se les ha asignado una textura o imagen y que siempre apuntan hacia la cámara, se usan para eliminar elementos secundarios que podrían llegar a ralentizar el sistema al ser calculados.

Las Optimizaciones de Cámara son aquellas que ayudan a mejorar el desempeño del sistema al no representar en pantalla, ni calcular en el proceso de síntesis de la imagen geometrías innecesarias. En esta categoría podemos encontrar técnicas de optimización tales como visibility groups, occlusion culling. La primera consiste en la definición de zonas de carga dentro de una aplicación tridimensional, a cada zona se le asigna un objeto que envuelve dicha zona de tal manera que solo una cámara que este dentro de dicho envoltente podrá visualizar los

objetos que se encuentran en esa zona, esto permite un aminoramiento del tiempo de carga y despliegue de modelos.

Por su parte occlusion culling (OC en adelante) se caracteriza por deshabilitar el renderizado de aquellos objetos que en un determinado momento no están siendo vistos por una cámara ya sea porque están fuera del ángulo de visión o están siendo tapados por otros objetos. (Unity3D, s.f.)

1.8.5 Recorridos Virtuales. Un recorrido virtual permite de forma remota acceder a un mundo virtual en el que se muestra un escenario real, por su parte (Ulldemolins, s.f.) define recorrido virtual como una simulación de un lugar virtual compuesto por una secuencia de imágenes, y recalca que esta definición puede quedarse corta, ya que actualmente y debido a los avances en la tecnología son muchas las formas que existen para dar a conocer un lugar a usuarios que no están presentes físicamente, entre estas forma se encuentra videos, panorámicas y sistemas de realidad virtual como Cave entre otras.

La forma en que se lleva al usuario la información que se quiere mostrar con el recorrido permite clasificarlos en dos categorías, recorridos fijos y recorridos interactivos.

Los primeros hacen referencia a aquellos en los que el usuario no interactúa con el entorno simplemente es un espectador que observa la secuencia que fue preparada por el creador del recorrido.

En cambio un recorrido interactivo, como bien lo dice su nombre, es aquel en el cual el usuario puede interactuar con el entorno, decidir dónde mirar, cuanto tiempo mirar, la experiencia va a depender del usuario debido a que se ofrece gran libertad.

Este proyecto se ubica en la categoría de los recorridos interactivos y a continuación se examinarán las tecnologías disponibles con las cuales es posible crear este tipo de recorrido.

1.8.6 Tecnologías disponibles. Actualmente se cuenta con gran cantidad de herramientas y tecnologías que hacen posible realizar recorridos muy cercanos a la realidad y que pueden brindar interactividad al usuario

1.8.6.1 Panorámico 360. Permite la visualización de un espacio con una vista panorámica de 360 grados, dando así la sensación de estar presentes en el medio del lugar y al girar permite visualizar todo lo que está alrededor.

Esta tecnología es conocida como panorama 360 cónico, consiste en realizar un mapeo de la imagen del entorno dentro de un cilindro, ver Figura 1

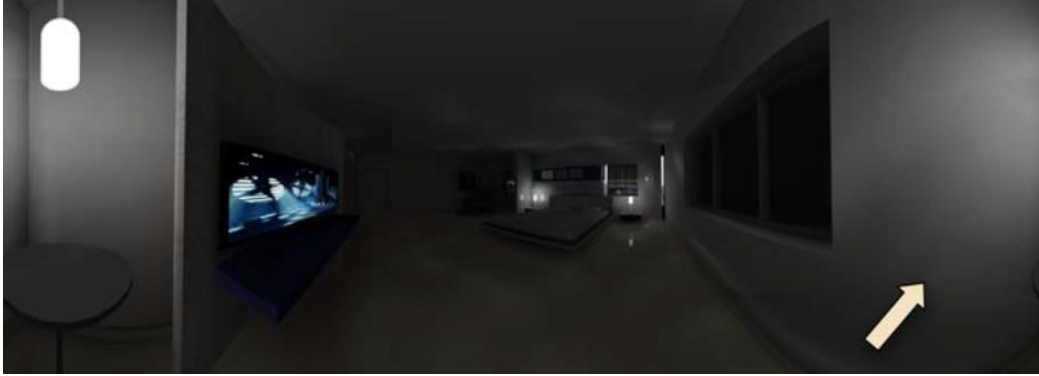


Figura 1. Mapeado cilíndrico.

Fuente: Ulldemolins. (s.f.), Mapeado cilíndrico, pág. 25

Una variación de esta tecnología es realizar el mapeo en una esfera, con lo que se conoce como panorama 360 esférico, ver Figura 2.



Figura 2. Mapeado esférico

Fuente: Ulldemolins. (s.f.), Mapeado esférico, pág. 25

La idea detrás de esta tecnología es tener un conjunto de imágenes consecutivas donde el final de una es el comienzo de la otra, dando así la sensación de continuidad. (Ulldemolins, s.f.)

1.8.6.2 Virtual Reality Modeling Language (VRML). VRML es un simple lenguaje de texto para describir formas 3D y entornos interactivos. (Nadeau, Moreland, & Heck, Michael M, s.f.), una escena está compuesta por objetos 3D que son descritos a través de formas geométricas básicas o que incluso por estructuras algo más complejas en las que especifican los vértices y aristas de cada polígono así como el color de su superficie. (Ulldemolins, s.f.), ver Figura 3.



Figura 3. Escena en formato VRML

Fuente: Ulldemolins. (s.f.), Escena en formato VRML, pág. 26

1.8.7 Herramientas de software empleadas en este proyecto

1.8.7.1 Blender. Fue concebido en diciembre de 1993 y apareció como un producto utilizable en agosto de 1994, integrando una serie de herramientas para la creación de una amplia gama de contenidos 2D y 3D. Blender ofrece un amplio espectro de funcionalidad para el modelado, texturizado, iluminación, animación y post-procesado de vídeo en un paquete. Por medio de su arquitectura abierta, ofrece interoperabilidad entre plataformas, extensibilidad, ocupa un espacio increíblemente pequeño en disco duro, y un flujo de trabajo altamente integrado. Blender es una de las aplicaciones de Código Abierto de gráficos 3D más populares del mundo.

Destinado a profesionales de medios y artistas del mundo, Blender puede usarse para crear visualizaciones 3D, tanto imágenes estáticas como videos de alta calidad, mientras que la incorporación de un motor 3D en tiempo real permite la creación de contenido interactivo que puede ser reproducido independiente de Blender.

Originalmente desarrollado por la compañía 'Not a Number' (NaN), Blender es ahora desarrollado como 'Software Libre', con el código fuente disponible bajo la licencia GNU GPL. La Blender Foundation, en los Países Bajos, es quién actualmente coordina su desarrollo. Entre el 2008 y el 2010, Blender ha sido enteramente reprogramado para mejorar sus funciones, flujo de trabajo e interfaz. El resultado de este trabajo es ahora conocido como Blender 2.6. (Blender, s.f.).

Dentro de sus principales características se pueden mencionar imagen procesada y pos-procesada; paquete de creación totalmente integrado, ofreciendo un amplio rango de herramientas esenciales para la creación de contenido 3D, incluyendo modelado, mapeado UV, texturizado, rigging, skinning, animación, simulaciones de partículas y de otros tipos, scripting, procesamiento, composición, pos-producción y creación de juegos; multiplataforma, con una interfaz basada en OpenGL, lista para ser usada en todas las versiones de Windows (98, NT, 2000, XP), Linux, OS X, FreeBSD, Irix, Sun y otros sistemas operativos. (Blender, s.f.)

1.8.7.2 Unity. Es un sistema de desarrollo de juegos multiplataforma con un potente motor de renderizado totalmente integrado con un juego completo de herramientas intuitivas y flujos de trabajo rápidos para crear contenido 3D interactivo; publicación multiplataforma fácil, miles de activos de calidad. (*Unity, s.f.*)

1.8.7.3 Gimp. Es un acrónimo para GNU Image Manipulation Program, es un programa de libre distribución para tareas tales como retocado de fotos, composición de imágenes y creación de imágenes, tiene muchas capacidades, de tal forma que puede ser usado como un simple programa de dibujo, un programa experto para retoque de imágenes de calidad, un sistema de procesamiento por lotes en línea, un generador de imágenes para la producción en masa, un convertidor de formato de imagen, un generador de imágenes para la producción en masa, un convertidor de formato de imagen.

GIMP es expandible y extensible. Está diseñado para ser ampliado mediante plugins y extensiones para hacer casi cualquier cosa. La interfaz de scripting avanzado permite todo, desde las tareas más simples a los procedimientos de manipulación de imágenes más complejas.

(GIMP, s.f.)

1.8.7.4 SketchUp Make. Como es descrito en su sitio web, es una herramienta gratuita para dibujo en 3D, que cuenta con una serie de herramientas útiles para facilitar el desarrollo de contenido. (*SketchUp, 2015*)

1.9 Marco Legal

La normatividad en la que se apoya la realización de este proyecto, se presenta a continuación.

1.9.1 Artículo 71 de la Constitución Política Colombiana. La búsqueda del conocimiento y la expresión artística son libres. Los planes de desarrollo económico y social incluirán el fomento a las ciencias y, en general, a la cultura. El Estado creará incentivos para personas e instituciones que desarrollen y fomenten la ciencia y la tecnología y las demás manifestaciones culturales y ofrecerá estímulos especiales a personas e instituciones que ejerzan estas actividades.

1.9.2 Ley 842 de 2003. Por la cual se modifica la reglamentación del ejercicio de la ingeniería, de sus profesiones afines y de sus profesiones auxiliares, se adopta el Código de Ética Profesional y se dictan otras disposiciones.

1.9.3 Licencias para Software Libre.

1.9.3.1 GNU General Public License. Una de las más utilizadas es la Licencia Pública General de GNU (GNU GPL). El autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL.

Es decir, la licencia GNU GPL posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia. Y añade que si se reutiliza en un mismo programa código "A" licenciado bajo licencia GNU GPL y código "B" licenciado bajo otro tipo de licencia libre, el código final "C", independientemente de la cantidad y calidad de cada uno de los códigos "A" y "B", debe estar bajo la licencia GNU GPL. (Free Software Foundation Inc, s.f.)

1.9.3.2 Creative Commons. Las licencias Creative Commons o CC están inspiradas en la licencia GPL de la Free Software Foundation. No son, sin embargo, un tipo de licenciamiento de software. La idea principal es posibilitar un modelo legal ayudado por herramientas informáticas, para así facilitar la distribución y el uso de contenidos.

1.9.4 Digital Millennium Copyright Act (DMCA). The Digital Millennium Copyright Act (DMCA) was signed into law by President Clinton on October 28, 1998. The legislation implements two 1996 World Intellectual Property Organization (WIPO) treaties: the WIPO Copyright Treaty and the WIPO Performances and Phonograms Treaty. The DMCA also addresses a number of other significant copyright-related issues. (U.S. Copyright Office Summary, 1998)

Diseño Metodológico

1.10 Tipo de Investigación

Este proyecto de investigación es de carácter descriptivo pues busca especificar propiedades, características y rasgos importantes de cualquier fenómeno que se analice (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2006, pág. 119), además tendrá un enfoque aplicado ya que según Tamayo (2004, p. 43) una investigación aplicada es el estudio y aplicación de la investigación a problemas concretos, en circunstancias y características concretas. Esta forma de investigación se dirige a su aplicación inmediata y no al desarrollo de teorías

1.11 Población

La población de la universidad Francisco de Paula Santander Ocaña, está constituida por 4685 estudiantes de los diferentes programas ofrecidos en ésta.

1.12 Muestra

De la población de la Universidad Francisco de Paula Santander Ocaña, se tomará una muestra representada por 355 estudiantes, para determinar el tamaño se esta se empleó la siguiente formula.

$$n = \frac{(Zc)^2 \cdot N \cdot p \cdot q}{(N - 1) \cdot E^2 + (Zc)^2 \cdot p \cdot q}$$

Dónde.

Tabla 1.
Descripción de variables.

Variable	Descripción
N	Tamaño de la población
Zc	Nivel de confianza (Zc=1.998)
p	Estimación de la proporción (P=0.5)
E	Error de estimación (d=0,05)

Fuente: Autores de Proyecto

Para una población N=4685

$$n = \frac{(1.998)^2(4685)(0.5)(0.5)}{(4685 - 1)(0.05)^2 + (1.998)^2(0.5)(0.5)}$$

$$n = \frac{4675.6346}{12.7080}$$

$$n = 367.928 \approx \approx n = 368$$

1.13 Técnicas e Instrumentos de Recolección de la Información

De acuerdo a las técnicas de recolección de información para una investigación aplicada, se ha optado por emplear la encuesta, para lo cual se han desarrollado un instrumento constituido por 11 preguntas de opción múltiple con única respuesta, dirigido a estudiantes, con el objetivo de conocer puntos de vista frente al problema y averiguar la aceptación que tendría el desarrollo de un recorrido virtual de la Universidad.

1.14 Análisis de la Información

En este trabajo la recolección de la información se realizó a través de la aplicación de una encuesta, en esta sección se muestran los resultados del análisis de la información obtenida.

La encuesta fue aplicada a 355 estudiantes de diferentes semestres de la Universidad Francisco de Paula Santander Ocaña, está constituida por 11 preguntas y tiene como objetivo determinar el nivel de aceptación por parte de los estudiantes de la Universidad frente a la creación de un recorrido virtual de esta.

A continuación se presentan los resultados obtenidos:

Tabla 2 .

Nivel de conocimiento que el estudiante tiene sobre la ubicación de la Universidad.

Variables	Frecuencia	Porcentaje
1	5	1,4%
2	15	3,9%
3	72	19,6%
4	129	35,1%
5	122	33,2%
NR	25	6,8%
Total	368	100,0%

Fuente: Autores de Proyecto.

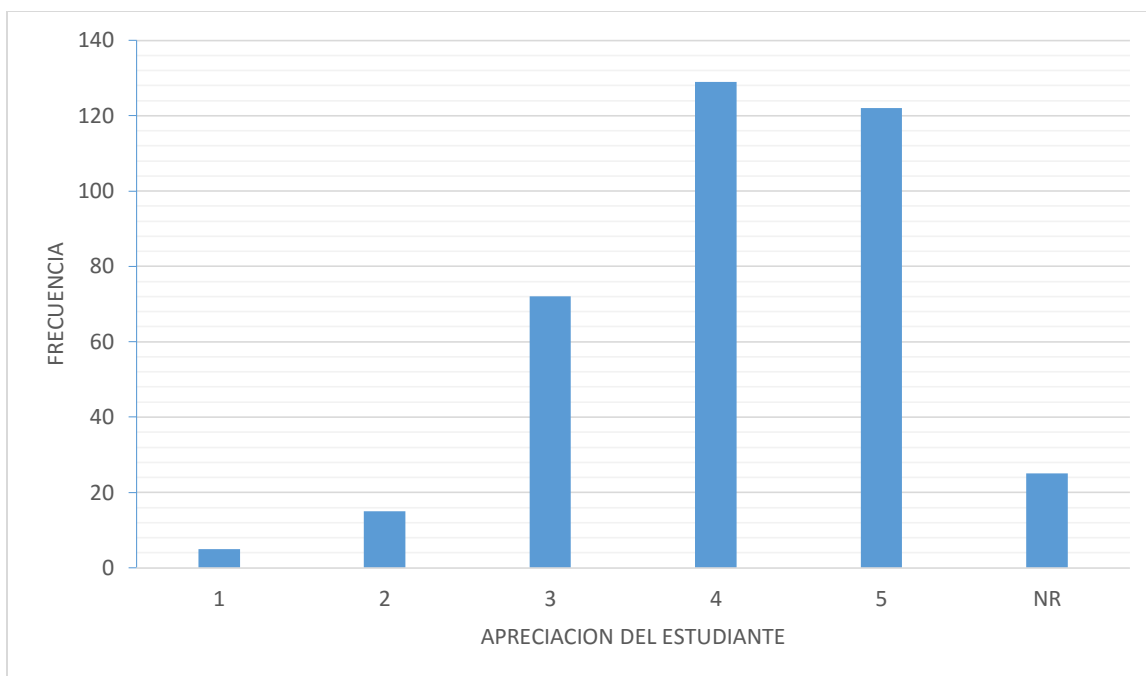


Figura 4. Nivel de conocimiento que el estudiante tiene sobre la ubicación de la Universidad.
Fuente: Autores de Proyecto.

Donde, 1 para muy malo, 2 es malo, 3 no bueno ni malo, 4 es bueno y 5 que es muy bueno.

De los estudiantes encuestados, el 35.05% considera que su conocimiento sobre la ubicación de la Universidad es bueno, el 33.15% cree que es muy bueno, el 19.57% cree que no es malo ni bueno, el 4.08% manifiesta que es malo frente al 1.36% el cual dice tener un conocimiento muy malo acerca de la ubicación de la UFPSO, mientras que el 6.79% de los encuestados no respondieron a esta pregunta.

Se observa que la gran mayoría de los encuestados conoce o tiene una noción acerca de la ubicación de la Universidad.

Tabla 3.*Nivel de dificultad que el estudiante ha tenido para ubicar oficinas en la Casona o Anexos.*

Variables	Frecuencia	Porcentaje
1	67	18,2%
2	49	13,,3%
3	101	27,4%
4	101	27,4%
5	44	12,0%
NR	6	1,6%
Total	368	100,0%

Fuente: Autores de Proyecto

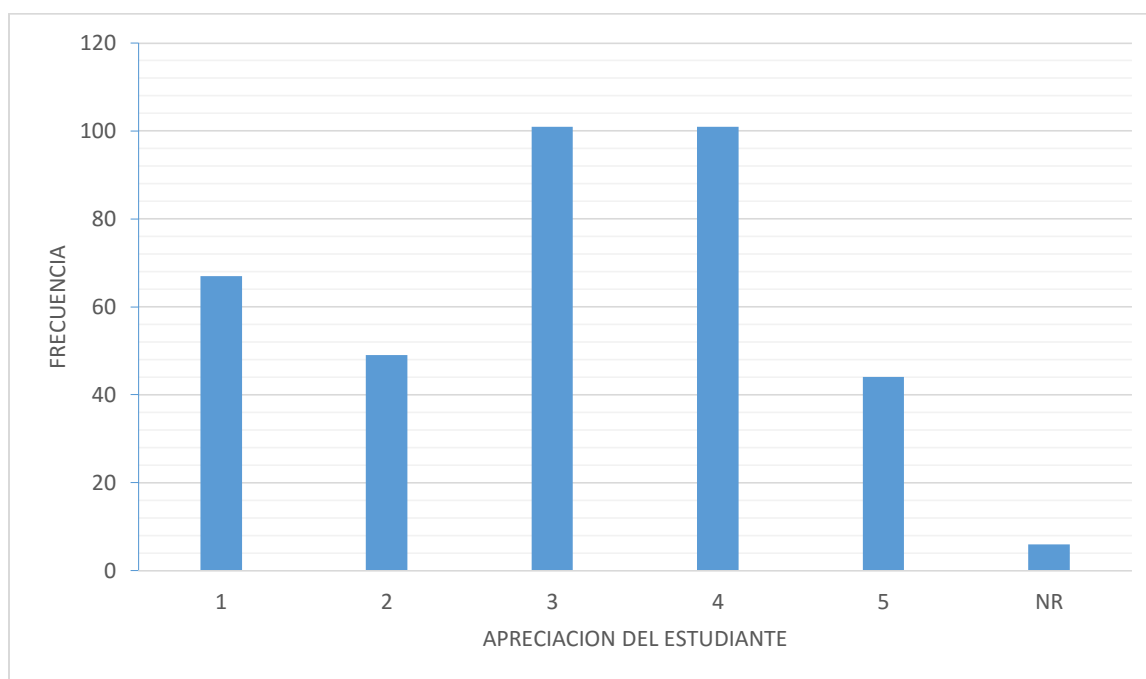


Figura 5. Nivel de dificultad que el estudiante ha tenido para ubicar oficinas en la Casona o Anexos.

Fuente: Autores de Proyecto.

Para esta pregunta se tiene como referencia la escala en la cual:

1 = muy poco; 2 = poco; 3 = ni poco ni mucho, 4 = mucho; 5 = demasiado

Del total de los estudiantes encuestados, el 27,45% de ellos manifiesta haber tenido ni pocos ni muchos problemas para encontrar dependencias en los edificios de Casona y Anexos, frente a otro 27.45 % según los cuales si han experimentado muchos problemas en dicho aspecto y un 11.96% que afirma que ha tenido demasiados problemas, un 18,21% dice haber tenido muy pocas dificultades y un 13,32% que expresa haber tenido pocas dificultades, mientras que el 1.63% de los que participaron en la encuesta no respondieron a esta pregunta.

Fijando como punto de referencia los que no han tenido ni pocas ni muchas dificultades para ubicar dependencias en dichos edificios de la UFPSO se observa que son más los que han tenido algunas o muchas dificultades frente a los que afirman no haber tenido o quienes dicen que han sido muy pocas.

Tabla 4.

Nivel de conocimiento que el estudiante tiene sobre las herramientas que la Universidad ofrece para facilitar la ubicación de las dependencias.

Variables	Frecuencia	Porcentaje
1	73	19,8%
2	100	27,2%
3	130	35,3%
4	49	13,3%
5	15	4,1%
NR	1	0,3%
Total	368	100,0%

Fuente: Autores de Proyecto.

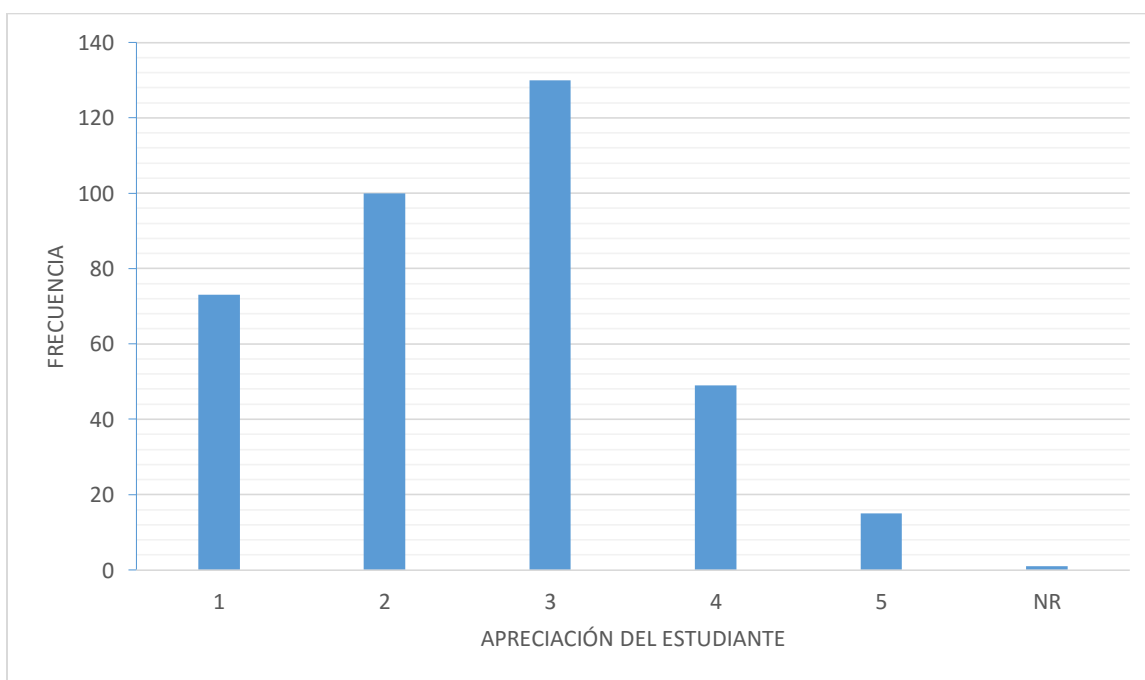


Figura 6. Nivel de conocimiento que el estudiante tiene sobre las herramientas que la Universidad ofrece para facilitar la ubicación de las dependencias.

Fuente: Autores de Proyecto.

De los estudiantes encuestados, el 35.3% consideran que su conocimiento sobre las herramientas que la UFPSO ofrece para facilitar la ubicación de las dependencias es intermedio, el 27.2% creen que es bajo y el 19.8% manifiestan que es muy bajo, frente al 13.3% los cuales lo consideran alto y el 4.1% quienes afirman tener un conocimiento muy alto sobre dichas herramientas.

Tabla 5.

Calificación que el estudiante le da a las herramientas a través de las cuales la universidad actualmente ofrece información de las dependencias.

VARIABLES	Frecuencia	Porcentaje
1	43	11,7%
2	86	23,4%
3	132	35,9%
4	77	20,9%
5	26	7,1%
NR	4	1,1%
Total	368	100,0%

Fuente: Autores de Proyecto.

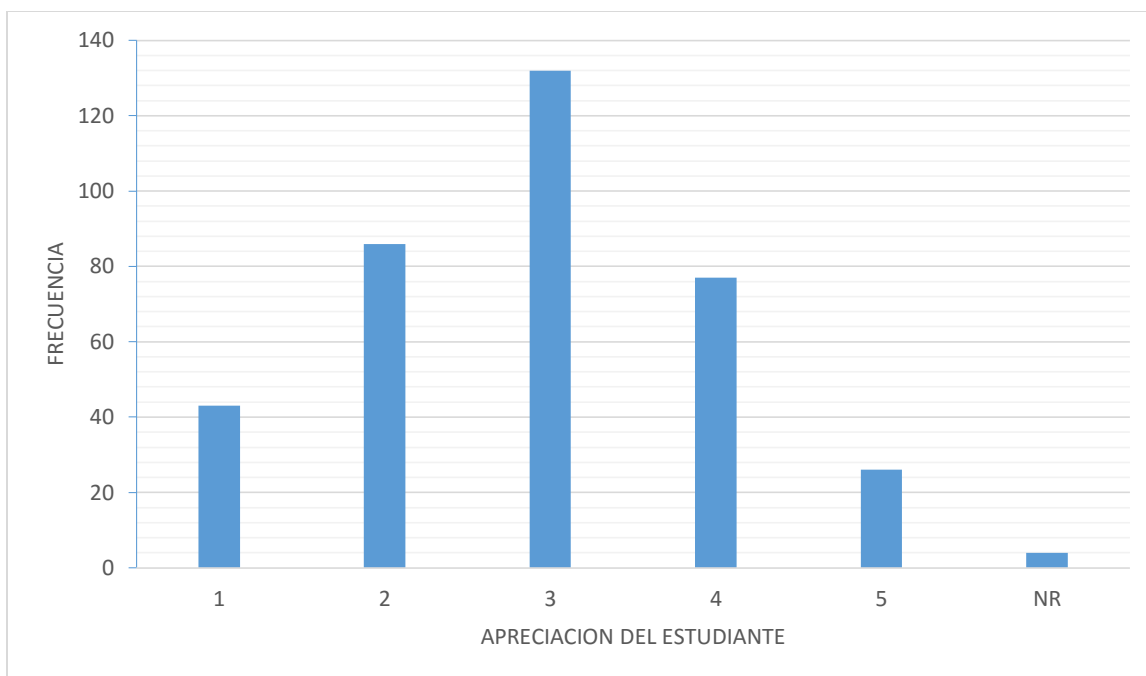


Figura 7. Calificación que el estudiante le da a las herramientas a través de las cuales la universidad actualmente ofrece información de las dependencias.

Fuente: Autores de Proyecto.

Del total de estudiantes encuestados, en una escala de 1 a 5, el 35.9% califican las herramientas actuales de información sobre la ubicación de las dependencias con 3, el 23.4% de los estudiantes le dan un 2 y el 11.7% las califican con un 1, frente al 20.9% los cuales les dan un 4 y el 7.1% que estiman la calificación de dichas herramientas en 5, el 1.1% de los encuestados no respondió a esta pregunta.

Tabla 6.
Frecuencia de utilización de los mapas de ubicación por parte del estudiante.

Variables	Frecuencia	Porcentaje
1	134	36,4%
2	86	23,4%
3	89	24,2%
4	38	10,3%
5	18	4,9%
NR	3	0,8%
Total	368	100,0%

Fuente: Autores de Proyecto.

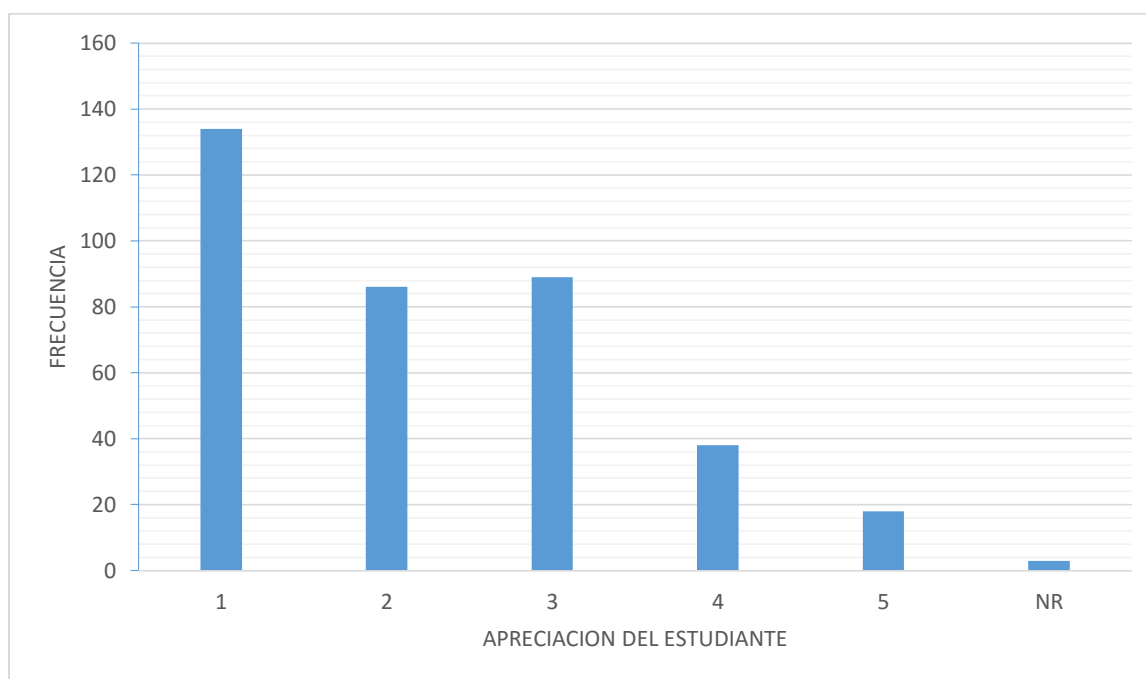


Figura 8. Frecuencia de utilización de los mapas de ubicación por parte del estudiante.
Fuente: Autores de Proyecto.

De los estudiantes encuestados, en cuanto a la frecuencia de utilización de los mapas de ubicación, el 36.4% manifiesta que los utiliza con muy poca frecuencia, el 23.4% dice utilizarlos con poca frecuencia y el 24.2% los cuales expresan que ni poco ni mucho, frente al 10.3% y 4.9% quienes aseguran utilizarlos con una alta y muy alta frecuencia respectivamente, el 0.8% de los encuestados no respondió.

Tabla 7.

Calificación por parte del estudiante, en cuanto a practicidad y facilidad de uso, de los mapas de ubicación de la Universidad.

VARIABLES	Frecuencia	Porcentaje
1	54	14,7%
2	74	20,1%
3	107	29,1%
4	92	25,0%
5	34	9,2%
NR	7	1,9%
Total	368	100,0%

Fuente: Autores de Proyecto.

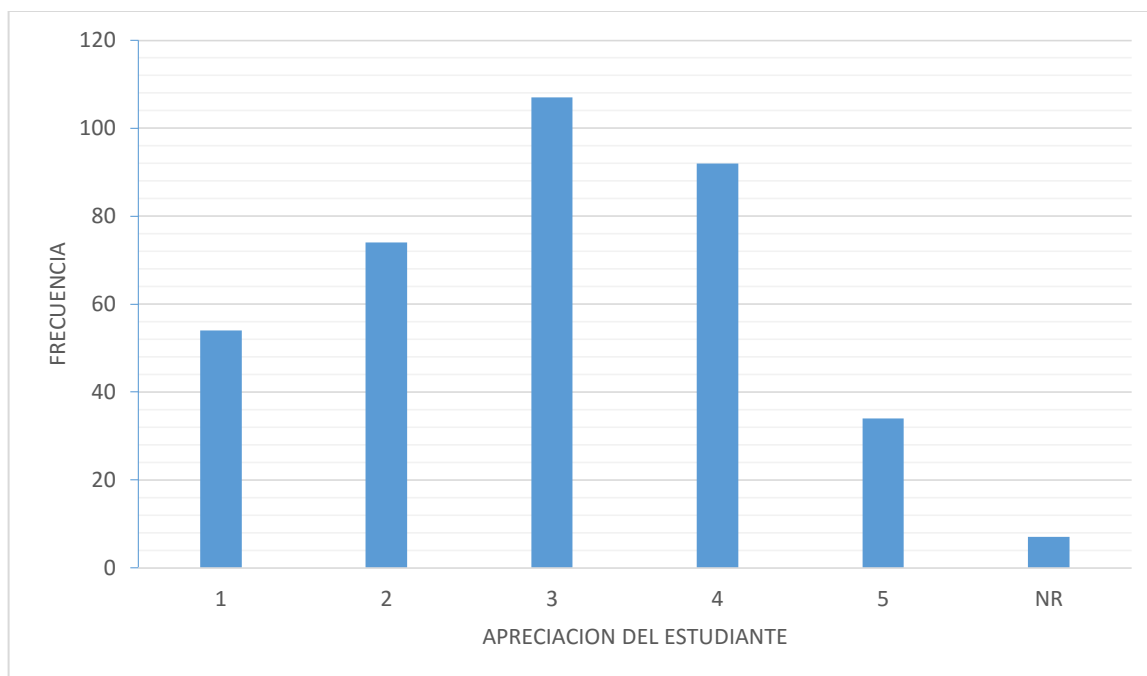


Figura 9. Calificación, en cuanto a practicidad y facilidad de uso, de los mapas de ubicación de la Universidad.
Fuente: Autores de Proyecto.

Del total de estudiantes encuestados, teniendo como criterios practicidad y facilidad de uso, el 29.1% califica con 3 los mapas de ubicación de la Universidad, el 25.0% los califican con 4 y el 9.2% les otorgan un 5, frente al 20.1% de los encuestados quienes les dan un 2 y el 14.7% que los califican con un 1, el 1.9% se abstuvo de responder a esta pregunta.

Tabla 8.

Nivel de satisfacción por parte del estudiante frente a la inducción ofrecida por la universidad en cuanto a organización del campus y sus dependencias.

Variables	Frecuencia	Porcentaje
1	39	10,6%
2	58	15,8%
3	73	19,8%
4	102	27,7%
5	93	25,3%
NR	3	0,8%
Total	368	100,0%

Fuente: Autores de Proyecto.

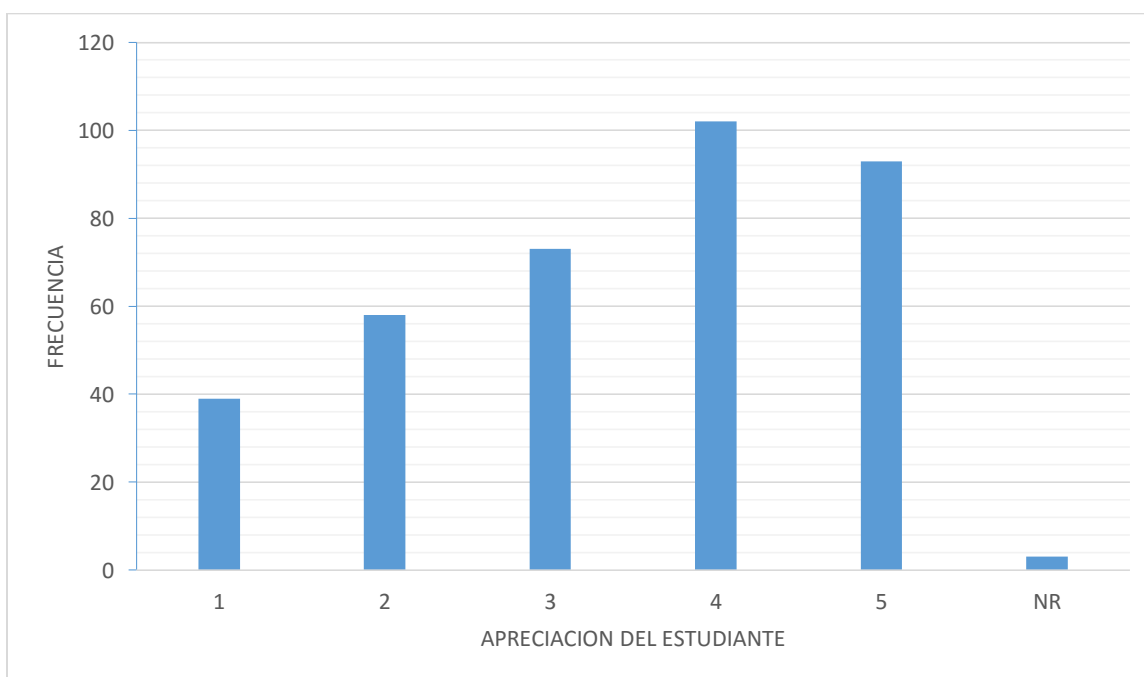


Figura 10. Nivel de satisfacción por parte del estudiante frente a la inducción ofrecida por la Universidad en cuanto a organización del campus y sus dependencias.

Fuente: Autores de Proyecto.

De los encuestados, frente a la pregunta que tan satisfechos se encuentran frente a la inducción ofrecida por la Universidad en cuanto a organización del campus y sus dependencias, el 19.8% manifiesta que su nivel de satisfacción es ni bajo ni alto, el 27.7% lo ubica en alto y el 25.3% en muy alto, frente al 15.8% que dice es bajo y 10.6% muy bajo, el 0.8% de los encuestados no respondió.

Tabla 9.

Nivel de satisfacción por parte del estudiante frente a la inducción ofrecida por la Universidad.

VARIABLES	Frecuencia	Porcentaje
1	22	6,0%
2	51	13,9%
3	112	30,4%
4	118	32,1%
5	60	16,3%
NR	5	1,4%
Total	368	100,0%

. Fuente: Autores de Proyecto

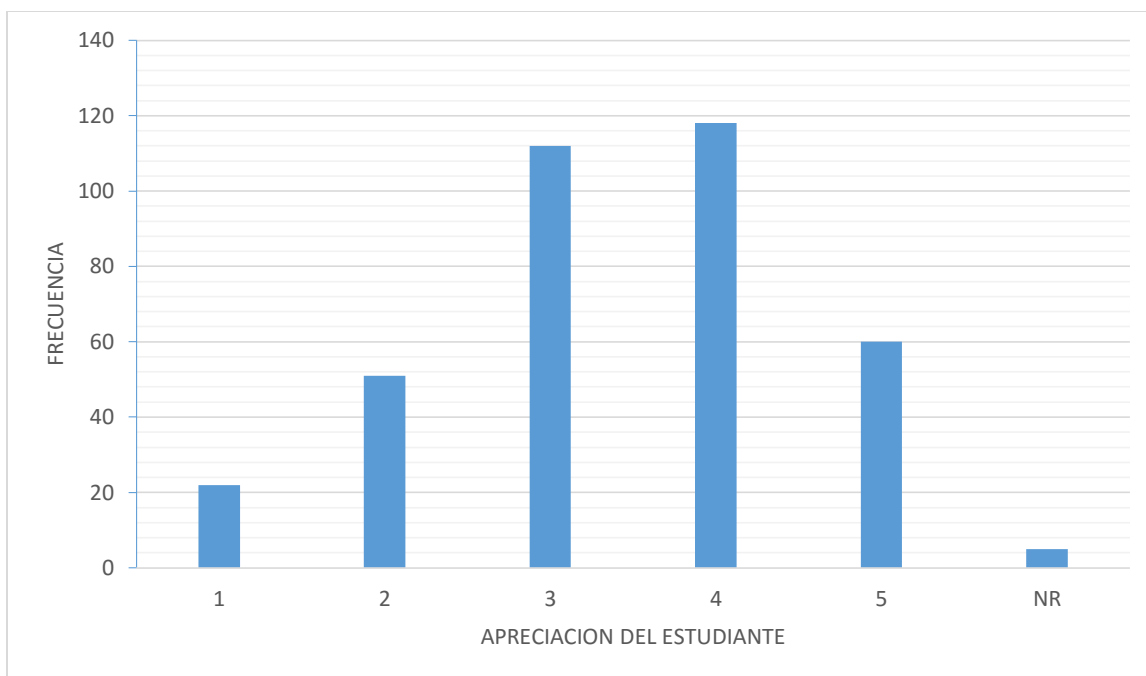


Figura 11. Nivel de satisfacción del estudiante respecto a la inducción ofrecida por la Universidad.
Fuente: Autores de Proyecto.

De los estudiantes encuestados, el nivel de satisfacción del 30.4% se encuentra en un nivel intermedio ni alto ni bajo, el 32.1% manifiesta que es alto y el 16.3% que es muy alto, frente al 13.9% y 6.0% los cuales afirman que su nivel de satisfacción es bajo y muy bajo respectivamente, el 1.4% de los encuestados no respondieron a esta pregunta.

Tabla 10.

Aceptación del estudiante frente a una herramienta accesible vía web que le permita conocer y recorrer los edificios Casona y Anexos de la UFPSO.

Variables	Frecuencia	Porcentaje
1	4	1,1%
2	18	4,9%
3	50	13,6%
4	110	29,9%
5	186	50,5%
NR	0	0,0%
Total	368	100,0%

Fuente: Autores de Proyecto.

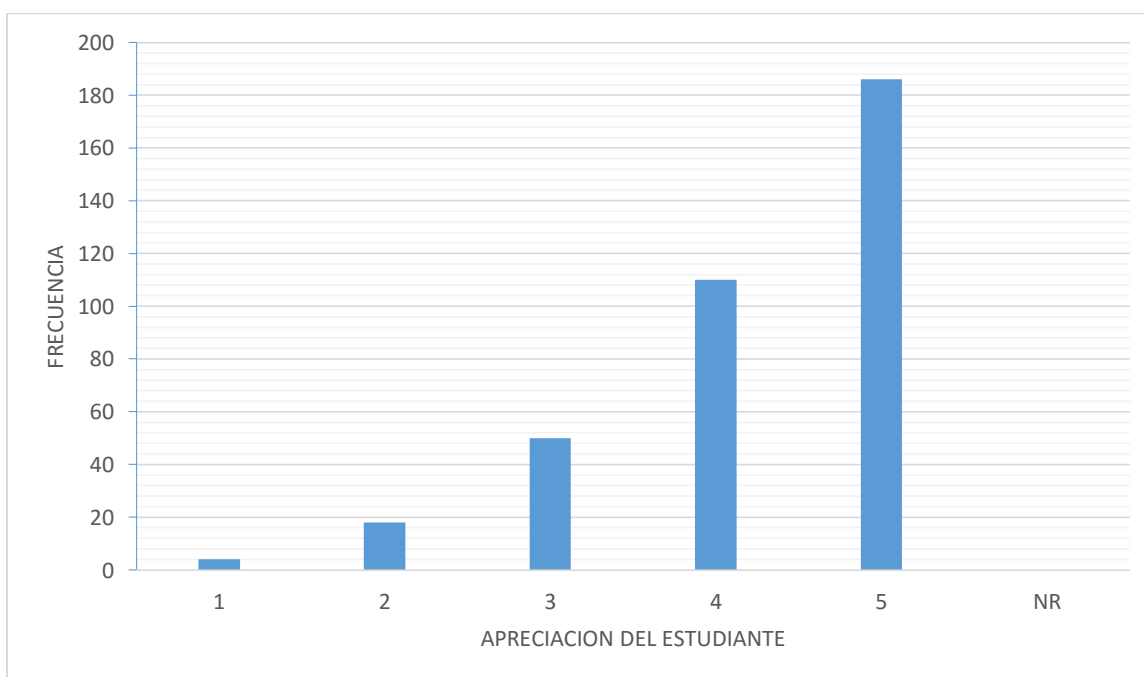


Figura 12. Aceptación del estudiante frente a una herramienta accesible vía web que le permita conocer y recorrer los edificios Casona y Anexos de la UFPSO.

Fuente: Autores de Proyecto.

Cuando se pidió que calificaran, en una escala de uno a cinco, la idea de ofrecer una herramienta accesible a través de internet que facilitara conocer los edificios de casona y anexos, de los estudiantes encuestados, el 50.5% calificó con 5 dicha idea, el 29.9% le dio un cuatro y el 13.6% le otorgó un 3, frente al 4.9% y 1.1% quienes dieron una calificación de dos y uno respectivamente frente a la idea expuesta anteriormente.

Tabla 11.
Nivel de interés del estudiante por dar a conocer la Universidad.

VARIABLES	Frecuencia	Porcentaje
1	5	1,4%
2	16	4,3%
3	58	15,8%
4	122	33,2%
5	165	44,8%
NR	2	0,5%
Total	368	100,0%

Fuente: Autores de Proyecto.

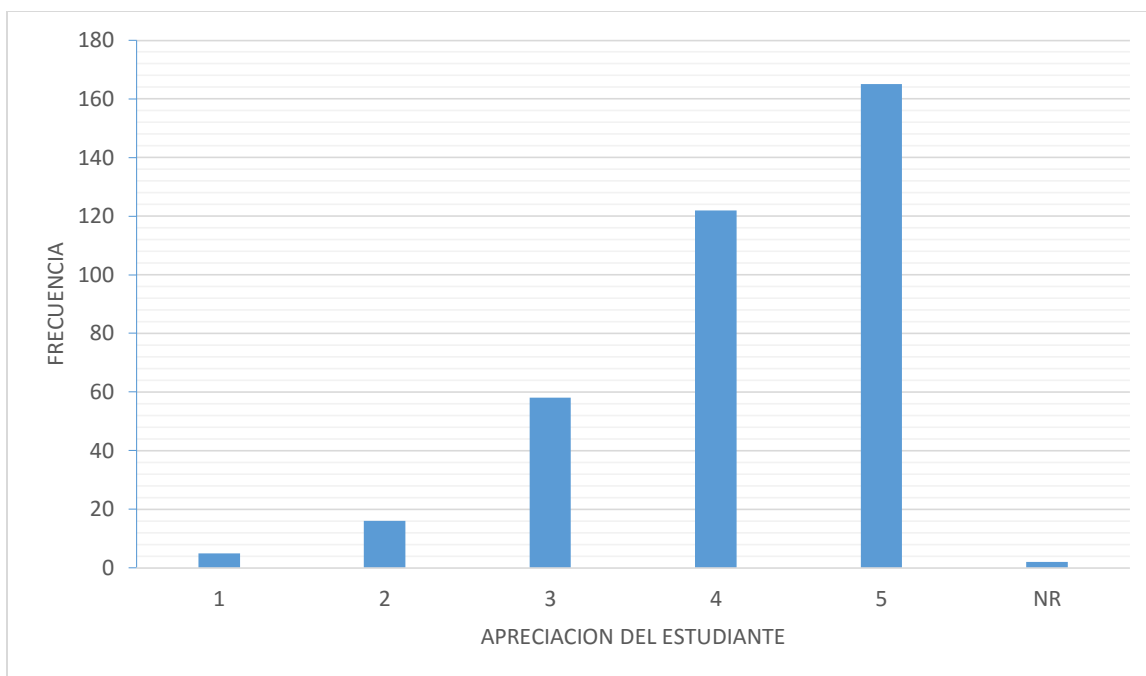


Figura 13 Nivel de interés del estudiante por dar a conocer la Universidad.
Fuente: Autores de Proyecto.

De los estudiantes encuestados, en una escala de uno a cinco, frente a la pregunta que tanto le gustaría poder mostrar el campus a personas que no lo conocen, el 44.8% respondieron con cinco, el 33.2% manifiesta una cuatro y el 15.8% un tres, frente al 4.3% y 1.4% quienes manifiestan un bajo interés en ese sentido con un dos y uno en dicha escala respectivamente.

Tabla 12.
Experiencias por parte del estudiante en cuanto a visitas virtuales a instituciones.

Variables	Frecuencia	Porcentaje
1	99	26,9%
2	72	19,6%
3	77	20,9%
4	69	18,8%
5	51	13,9%
NR	0	0,0%
Total	368	100,0%

Fuente: Autores de Proyecto

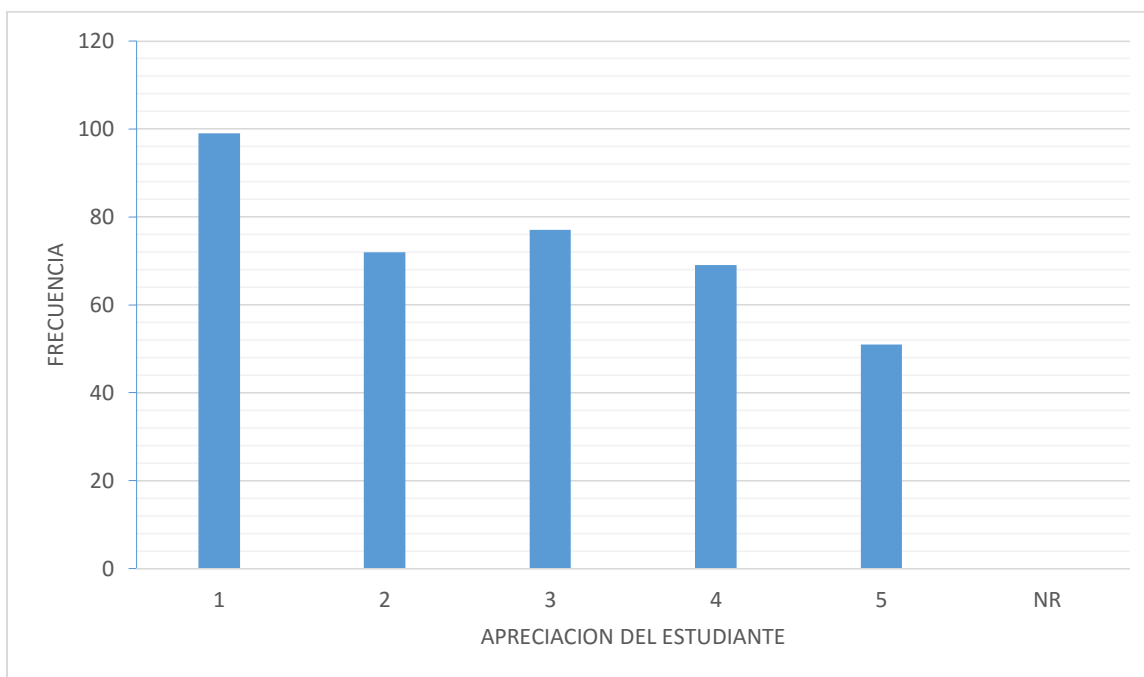


Figura 14. Experiencias por parte del estudiante en cuanto a visitas virtuales a instituciones.

Fuente: Autores de Proyecto

De los estudiantes encuestados, en cuanto a experiencias por visitas virtuales, el 26.9% manifiesta nunca haber realizado alguna, el 19.6% dice que muy pocas veces y el 20.9% que expresa que algunas veces, frente al 18.8% y el 13.9% quienes manifiestan haber experimentado bastantes experiencias de este tipo.

1.15 Metodología de Desarrollo.

El objetivo de este proyecto es brindar un recorrido virtual del campus universitario, con el fin de dar a conocer y facilitar el reconocimiento de los edificios Casona y Anexos de la universidad Francisco de Paula Santander seccional Ocaña; para el desarrollo de este se plantean las siguientes fases en base a los objetivos establecidos previamente.

Primer objetivo. Modelar estructuras y objetos de los edificios “La Casona” y “Anexos”, así como el personaje guía del recorrido, con el fin de tener los prototipos en 3D.

Fase uno, se recopilan los planos de la universidad, material fotográfico y medición de objetos (puertas, ventanas, mesas, etc.) de las dependencias que se van a incluir en el recorrido virtual, necesarios para obtener una mejor comprensión de organización del entorno.

Fase dos, se realiza esculpido de los modelos de los edificios, casona y anexos, así como el de las vías de acceso y objetos que integran los mismos. De igual manera se realiza el esculpido del personaje institucional Santanderito y se hace el Rigging del personaje.

Fase tres, se realiza el proceso de texturizado, separando los diferentes materiales que componen cada objeto, creando sus respectivo mapeado UV si es pertinente y se agregan los materiales a cada componente del objeto.

Segundo objetivo. Organizar el entorno del recorrido virtual con el fin de integrar y dar interactividad a los objetos creados.

Fase uno, se obtienen un prototipo del terreno del campus que se va a utilizar mediante SketchUp, para posteriormente realizar ajustes más detallados de éste en Blender y posteriormente en el editor de Unity 3D.

Fase dos, se importan los modelos creados previamente como Asset en el motor gráfico ubicándolos en el terreno generado dando una impresión realista del entorno.

Fase tres, con el fin de optimizar el rendimiento y dar cierta interactividad con el entorno se crearon unos scripts que permitan realizar estas operaciones junto con el proceso de lighting, Ambient Oclusion y sombras que darán un efecto más realista a la escena.

Fase cuatro, para mejor ambientación del recorrido se planea que Santanderito haga una introducción, para lo cual se elabora un dialogo; después de esto se realiza una sincronización labial que permita una adaptación entre las voces y los movimientos vocales del personaje.

Fase cinco, con el fin de dar una mejor presentación al aplicativo se realizó una pantalla de inicio que contiene diferentes opciones de configuración y un menú de fácil navegación.

Tercer objetivo, Incorporar la aplicación desarrollada del recorrido virtual en un sitio web, facilitando el acceso por medio de internet.

Fase uno, se ajustan los parámetros de compilación, calidad y efectos, generando así los archivos del aplicativo que permitirán su ejecución a través de un sitio web.

Fase dos, se suben los archivos obtenidos de la compilación a un subdominio hospedado en los servidores de la Universidad Francisco de Paula Santander Ocaña.

Presentación de resultados

La presentación de los resultados obtenidos con el desarrollo del presente proyecto se estructuró de acuerdo a las fases establecidas anteriormente.

1.16 Modelar estructuras y objetos de los edificios “La Casona” y “Anexos”, así como el personaje guía del recorrido, con el fin de tener los prototipos en 3D.

1.16.1 Recopilación de planos y material fotográfico. Esta fase comprende la recopilación de planos del campus universitario, en específico de los edificios Casona, Anexos y las vías de acceso a los mismos; para ello se recurrió a la división de Planeación y se solicitó al Arquitecto dicho material, el cual fue suministrado en archivos de AutoCad, ver Apéndice A. De igual manera se tomaron fotos de cada una de las dependencias y los elementos que componen cada oficina.

1.16.2 Modelado en Blender. Durante esta fase se crearon los modelos virtuales de Casona, Anexos, los caminos que la interconectan y las diferentes divisiones y objetos que componen cada uno de estos lugares; para tal motivo se realizó mediante los siguientes pasos:

1.16.2.1 Importación del Croquis. Este proceso se realizó con la ayuda de un script desarrollado para Blender en su versión 2.48, el cual permite tomar los planos, ver Apéndice B, en formato DXF (acrónimo del inglés Drawing Exchange Format), el cual es generado por AutoCad, convirtiéndolos en una malla trabajable en Blender. Para ello se empieza accediendo al menú File, en la opción Import y se selecciona Autodesk DXF, ver Figura 15.

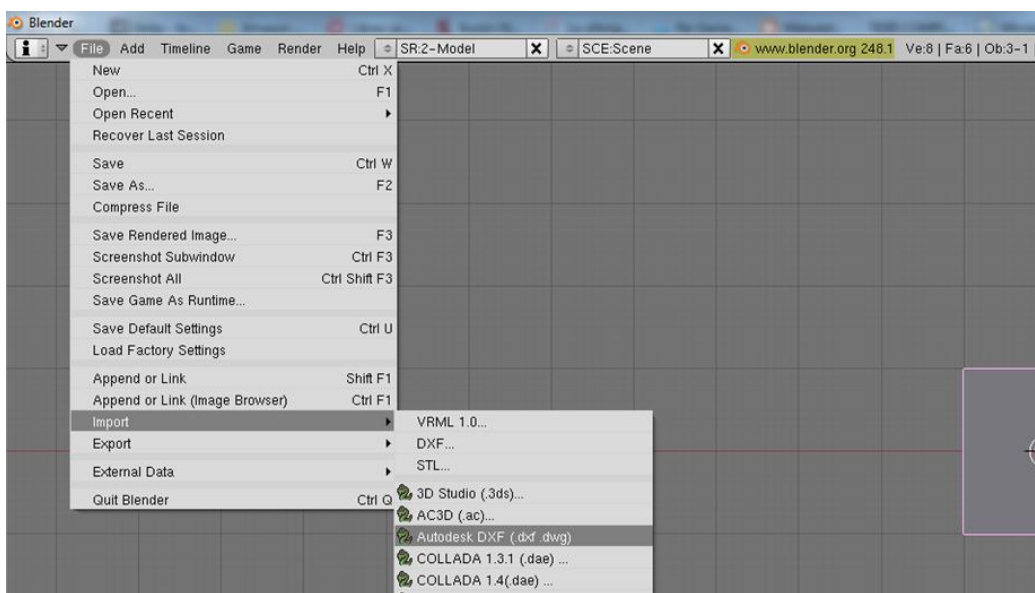


Figura 15. Importar archivo DXF
Fuente: Autores del proyecto

Posteriormente se selecciona el archivo a importar y se presiona el botón START IMPORT ver Figura 16, de lo cual se obtuvo una malla con la estructura de los edificios como se puede ver en la Figura 17.

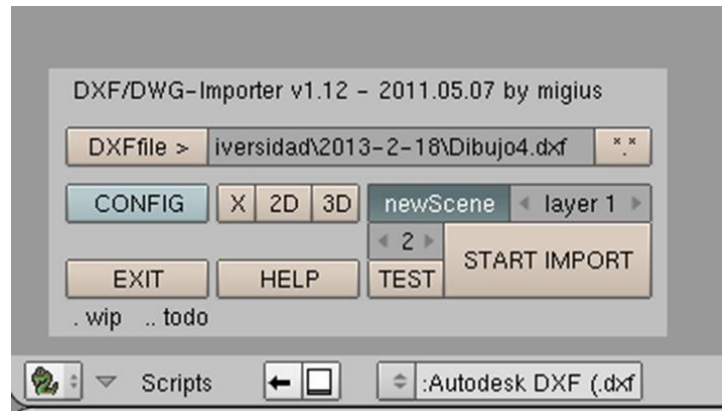


Figura 16. Importar Autodesk DXF
Fuente: Autores del proyecto

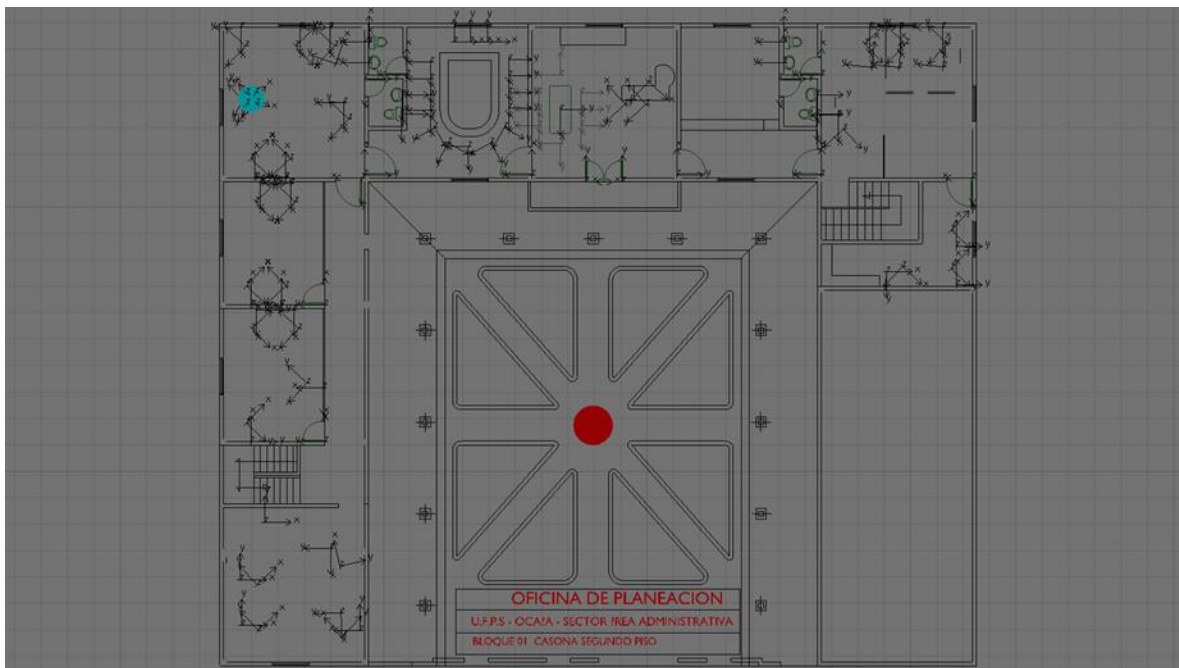


Figura 17. Plano importado
Fuente: Autores del proyecto

Posteriormente se pasaron las mallas obtenidas a una versión de Blender más actualizada para acceder a las mejoras y opciones recientes de este programa, esto debido a un cambio drástico en funcionalidad e interfaz que obtuvo este software después de la versión 2.6; se trabajó en la versión 2.48a pues es la versión funcional del script.

1.16.2.2 Modelado de edificios. Una vez teniendo las mallas extraídas de los planos correspondientes a los edificios y vías, procedemos a eliminar objetos que permanecen en la malla y no son útiles para el desarrollo; así mismo se organiza la malla debido a errores y bordes duplicados y superpuestos ver Figura 18.

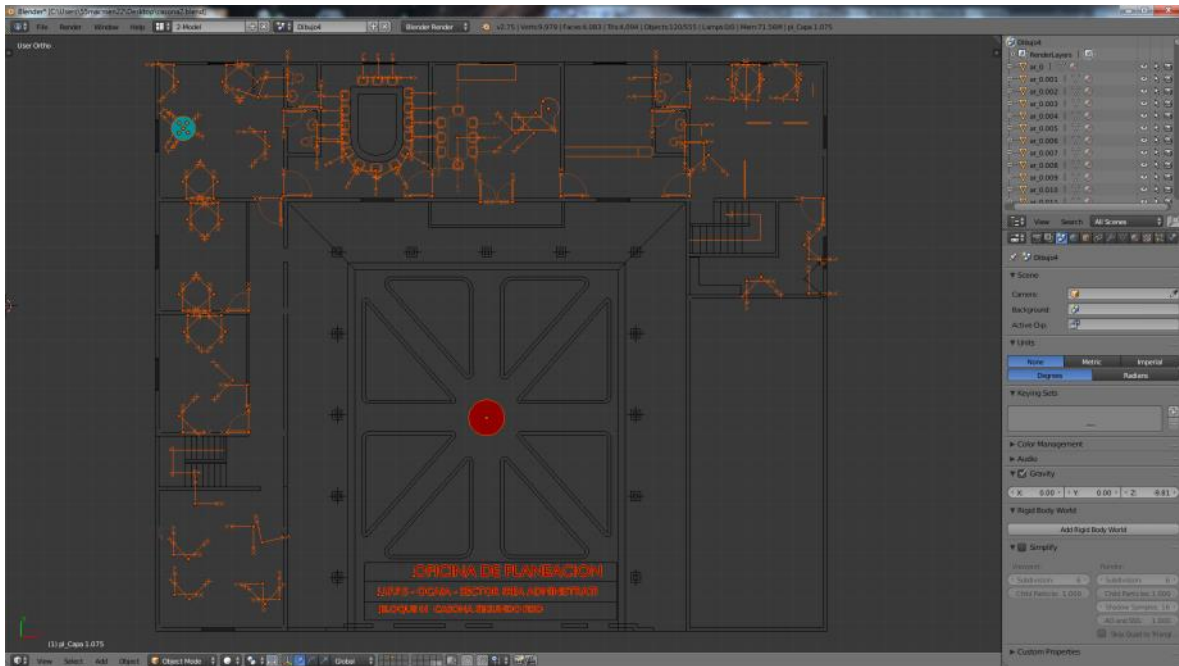


Figura 18. Mejoras a la malla
Fuente: Autores del proyecto

Mediante manipulación de la malla se dio forma a las paredes, techos, vigas y escaleras que conforman la estructura de los edificios, obteniendo de esta manera un modelo relista de los mismos, ver Figura 19 y Figura 20.

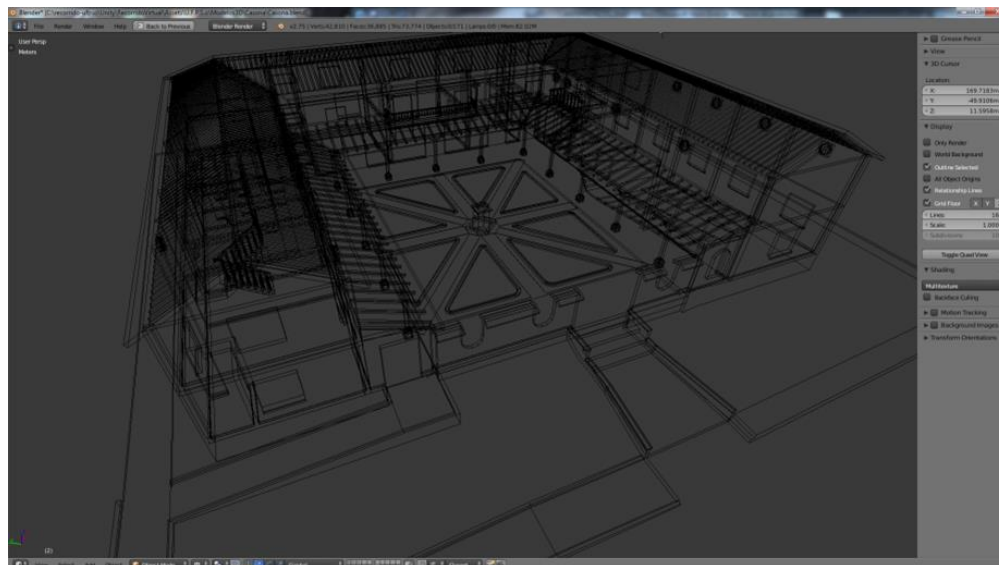


Figura 19. Vista de alambres, edificio Casona
Fuente: Autores del proyecto

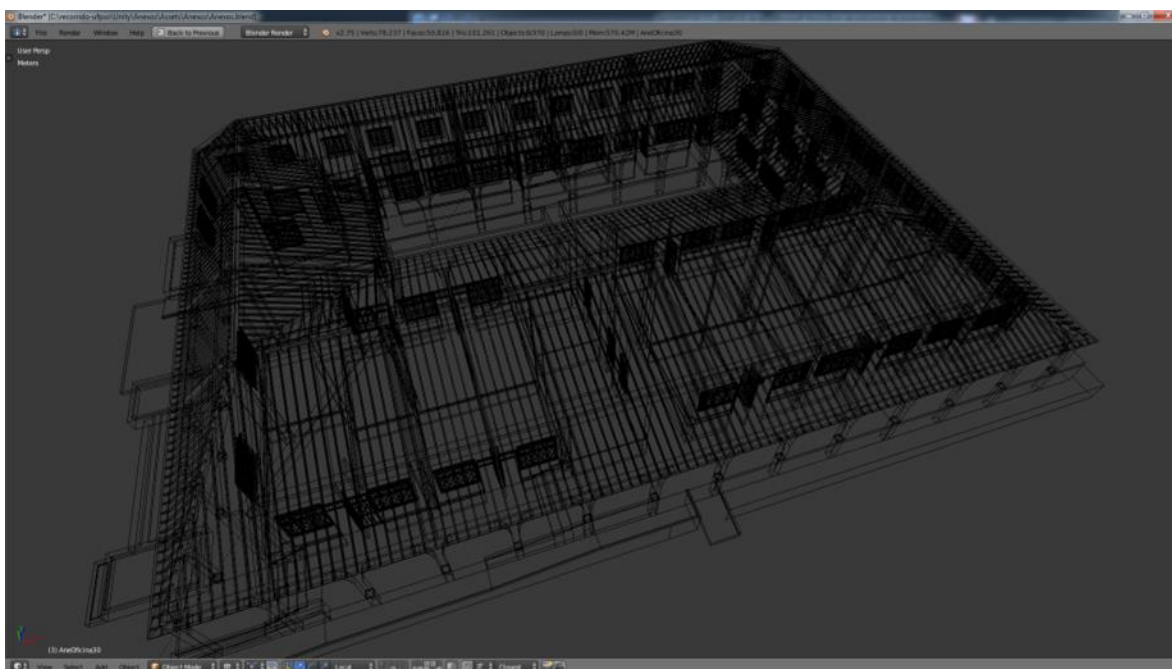


Figura 20. Vista de alambres, edificio Anexos Académicos
Fuente: Autores del proyecto

Para el modelo de las vías se ajustaron los niveles de altura de acuerdo a las curvas de nivel provenientes de los planos que fueron proporcionados por la universidad, para dar una sensación de profundidad y perspectiva acorde a la realidad, además de esto como en los modelos anteriores su forma y distribución se basó en la malla importada de AutoCad. Adicionalmente se incluyó una vista básica de los demás edificios y caminos que conforman la parte académica y con mayor concurrencia del campus, esto con dos objetivos, dar una mejor vista y facilitar el desarrollo de próximas versiones de este proyecto, ver Figura 21.

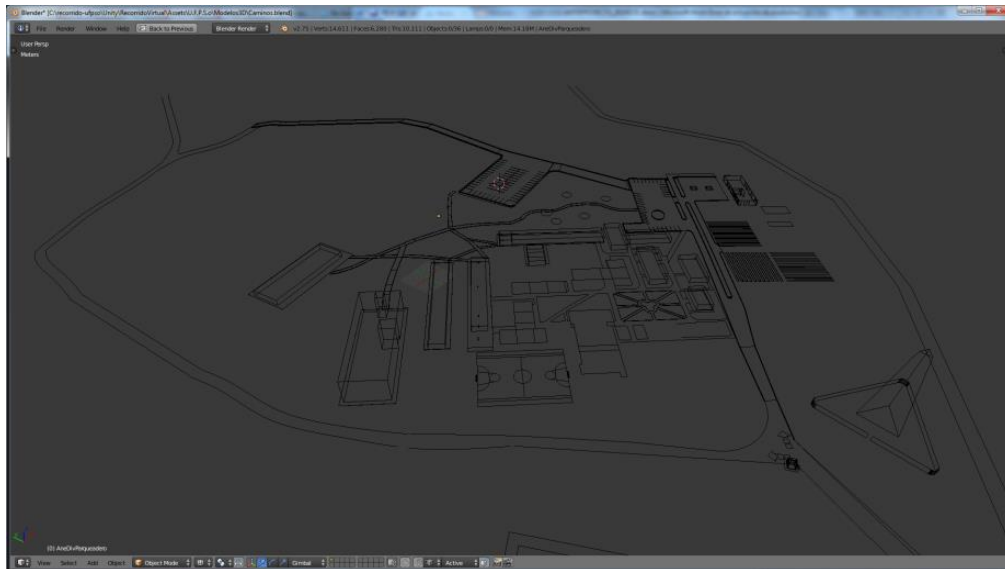


Figura 21. Vista de alambres, caminos y edificios de clases de la Universidad Francisco de Paula Santander Ocaña.
Fuente: Autores del proyecto

El modelado de los objetos y divisiones que conforman los edificios se realizó en base a las imágenes tomadas y algunas medidas complementarias, empleando la técnica de Box Modeling, la cual consiste en dar forma a los objetos a partir de una caja, utilizando las herramientas de extrusión, subdivisión, deformación (arrastre de vértices), escalado, entre otros. Ver Figura 22.

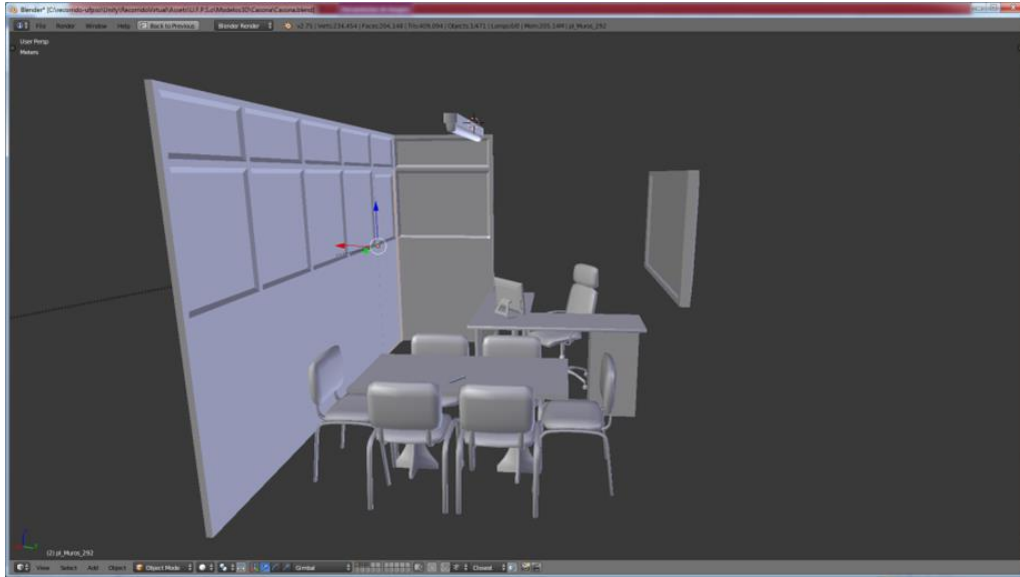


Figura 22. Modelado de objetos y divisiones.
Fuente: Autores del proyecto

1.16.2.3 Modelado y Rigging del personaje. En base a los modelos en 2D originales del personaje institucional Santanderito, se crearon sus respectivas vistas: frontal, lateral y posterior, ver Figura 23, con las cuales se inició el modelado del personaje utilizando la técnica de Box Modeling con el fin de ajustar la forma al modelo obtenido en las vistas, hasta obtener una aproximación como se muestra en la imagen, ver Figura 24.

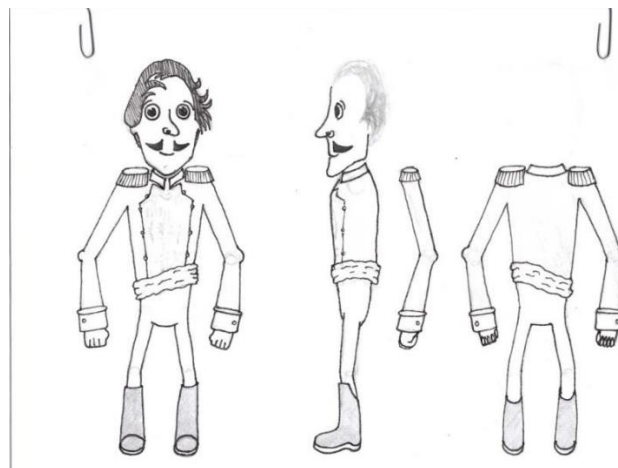


Figura 23. Vistas del personaje Santanderito
Fuente: Autores del proyecto



Figura 24. Modelo 3d del personaje Santanderito
Fuente: Autores del proyecto

Para el proceso de Rigging, se agregan los huesos, ver Figura 25, que permitirán al personaje adoptar posiciones y animaciones, cada hueso en específico tiene un área de influencia, la cual se calcula automáticamente, pero en ocasiones es necesario corregir estas influencias debido a cambios drásticos en la malla que dan un comportamiento poco natural. Para el Rigging facial, se seleccionan los vértices que conforman los grupos claves, como los labios, mentón bajo, mejillas, entre otros; una vez creados los grupos de vértices, se procede a crear las formas claves en base a los movimientos vocales y faciales necesarios, ver Figura 26.



Figura 25. Vista de esqueleto.
Fuente: Autores del proyecto

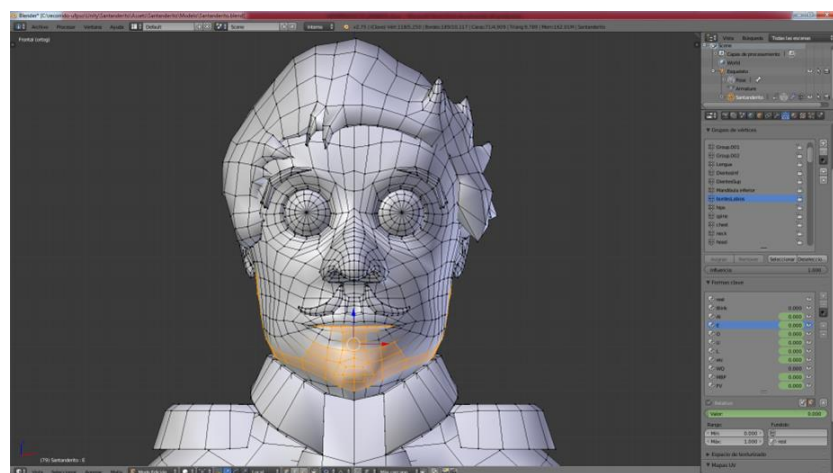


Figura 26. Grupos de vértices y formas clave.
Fuente: Autores del proyecto

1.16.3 Texturizado. En proceso de texturizado se realizó de dos formas; para el personaje Santanderito se creó un solo mapa de UV completo en el cual se agregó una imagen con los colores y materiales del personaje, ver Figura 27 . Para los demás objetos como paredes, mesas, televisores, sillas, entre otros, se creó el mapa UV solo en las secciones que llevasen alguna textura junto con el material, y en las otras secciones solo se agregó el material necesario, ver Figura 28.

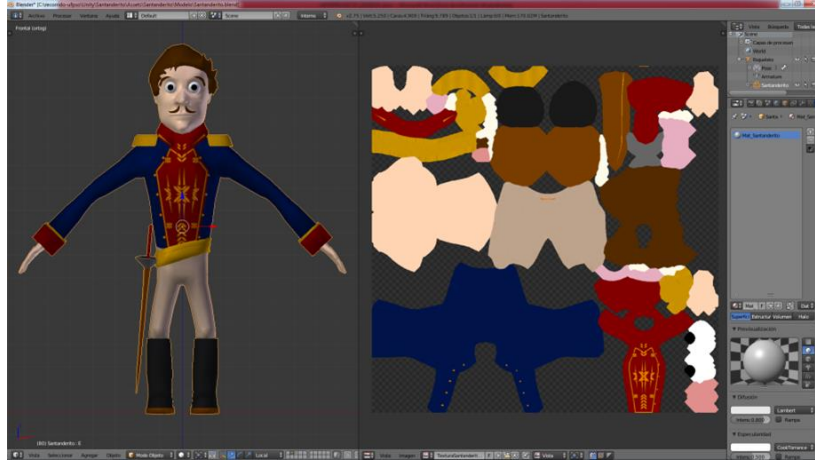


Figura 27. Textura Santanderito
Fuente: Autores del proyecto

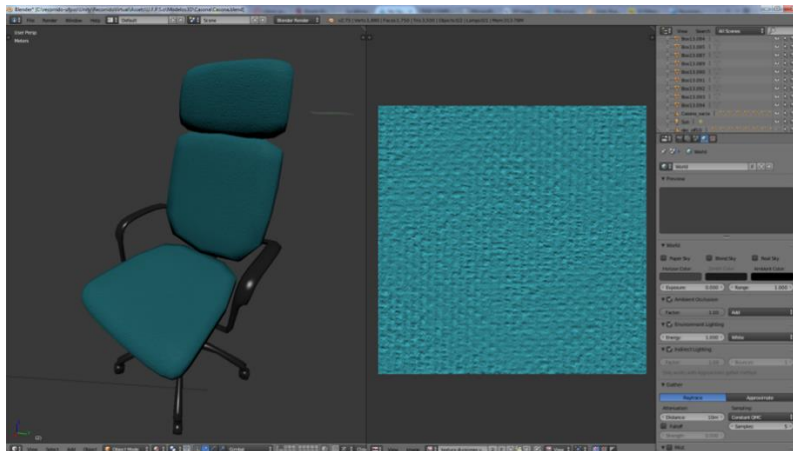


Figura 28. Silla con texturas
Fuente: Autores del proyecto

1.17 Organizar el entorno del recorrido virtual con el fin de integrar y dar interactividad a los objetos creados.

Para dar la posibilidad de recorrer los lugares creados previamente, se optó por usar la herramienta Unity 3D como motor gráfico, ya que este cuenta con una gran documentación

accesible por diversos medios y fácil integración con los navegadores web gracias a su plugin o mediante el uso de HTML 5.

La integración de todos los componentes del recorrido se llevó a cabo de la siguiente manera, inicialmente se creó el terreno, el siguiente paso fue la importación de cada uno de los modelos 3D creados en la fase de modelado, es decir, los edificios Casona y Anexos Académicos y cada uno de los objetos que hay dentro de las oficinas y finalmente se agregaron los árboles y demás vegetación con el fin de dar una mejor vista de la escena.

Luego de se llevaron a cabo optimizaciones para mejorar el rendimiento del recorrido, también se desarrollaron las diferentes escenas que constituirían el recorrido.

1.17.1 Creación de terreno. En la etapa de creación del terreno se partió de la información suministrada en el plano general de la UFPSO sobre la ubicación de las diferentes zonas y edificios. Comenzando con la simplificación del plano general de La UFPSO, en la cual se ocultaron detalles que no eran relevantes en la fase de generación del terreno, posteriormente se guardó el plano optimizado como un archivo de extensión dxf. El siguiente paso fue importarlo en Blender y convertido en un modelo 3D, ver Figura 29. Vista superior en Blender del Plano general simplificado de la Universidad., llevando a cabo el mismo proceso descrito en la sección 4.2.1.1 de este documento.

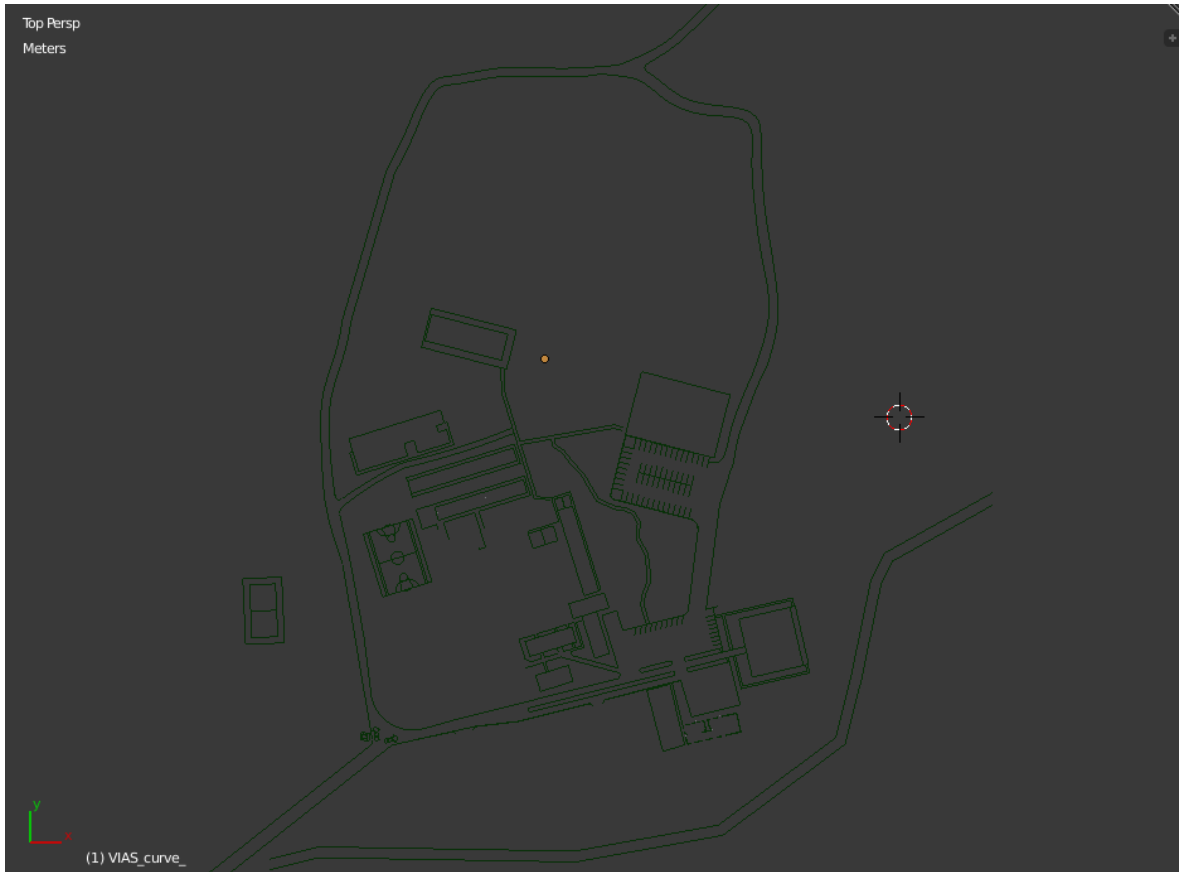


Figura 29. Vista superior en Blender del Plano general simplificado de la Universidad.
Fuente: Autores del proyecto

Ya que se buscaba crear un terreno lo suficientemente amplio de tal forma que se lograra captar detalles de los alrededores con el objetivo de hacer el terreno más cercano al real, era necesario que además de la información que se obtuvo del plano de las zonas cercanas a la Universidad, conseguir información sobre las alturas y ubicación de las montañas de los alrededores. Para eso se optó por usar el programa SketchUp, que a través de la opción ubicada en la ruta Archivo/Geolocalización/Añadir Localización, despliega una interfaz en la que es posible buscar una región o lugar por su nombre y posteriormente seleccionar la zona deseada, ver Figura 30.

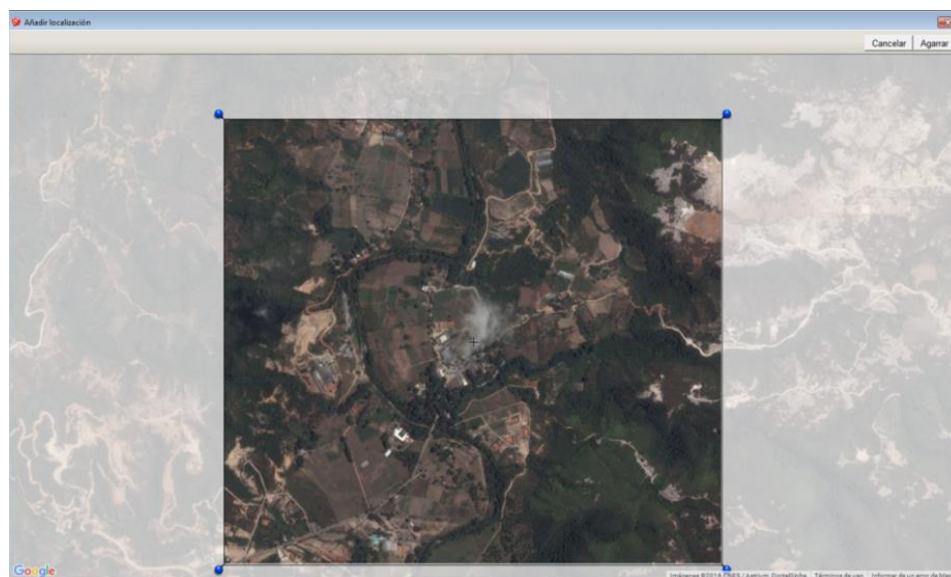


Figura 30. Selección del terreno alrededor de la UFPSO en el programa SketchUp.
Fuente: Autores del proyecto

Luego de que se selecciona la zona se pulsa el botón “Agarrar” ubicado en la parte superior derecha, el resultado es un plano ubicado en el espacio tridimensional (x, y, z) con la imagen de la zona seleccionada, ver Figura 31.

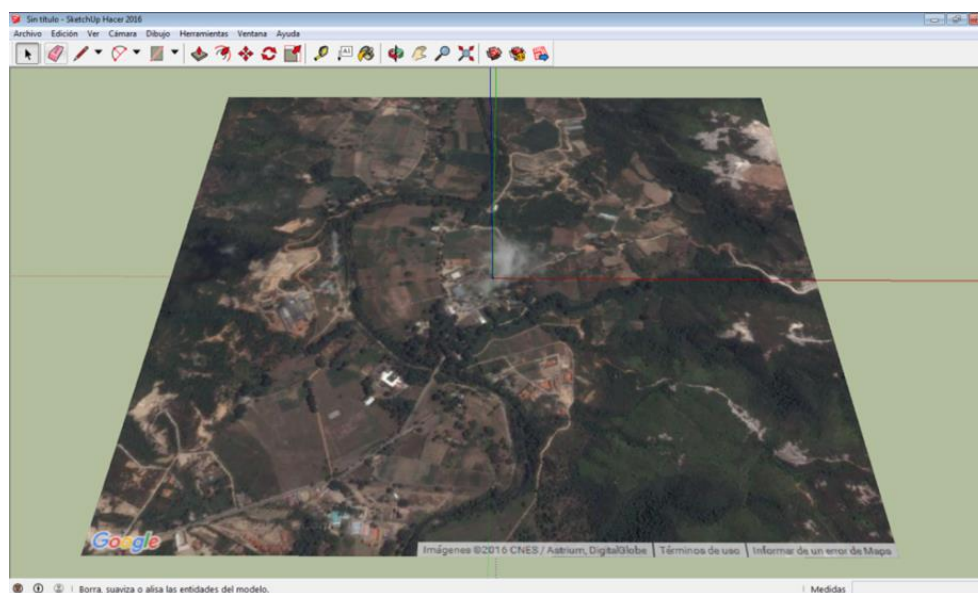


Figura 31. Plano generado.
Fuente: Autores del proyecto

Como se aprecia en la Figura 31, el plano no muestra ninguna elevación, para poder visualizar esto es necesaria la ejecución del comando ubicado en la ruta “Archivo/Geolocalización/Mostrar terreno”, lo que da como resultado que el plano inicial se deforme de acuerdo a la geografía de la zona ver Figura 32.

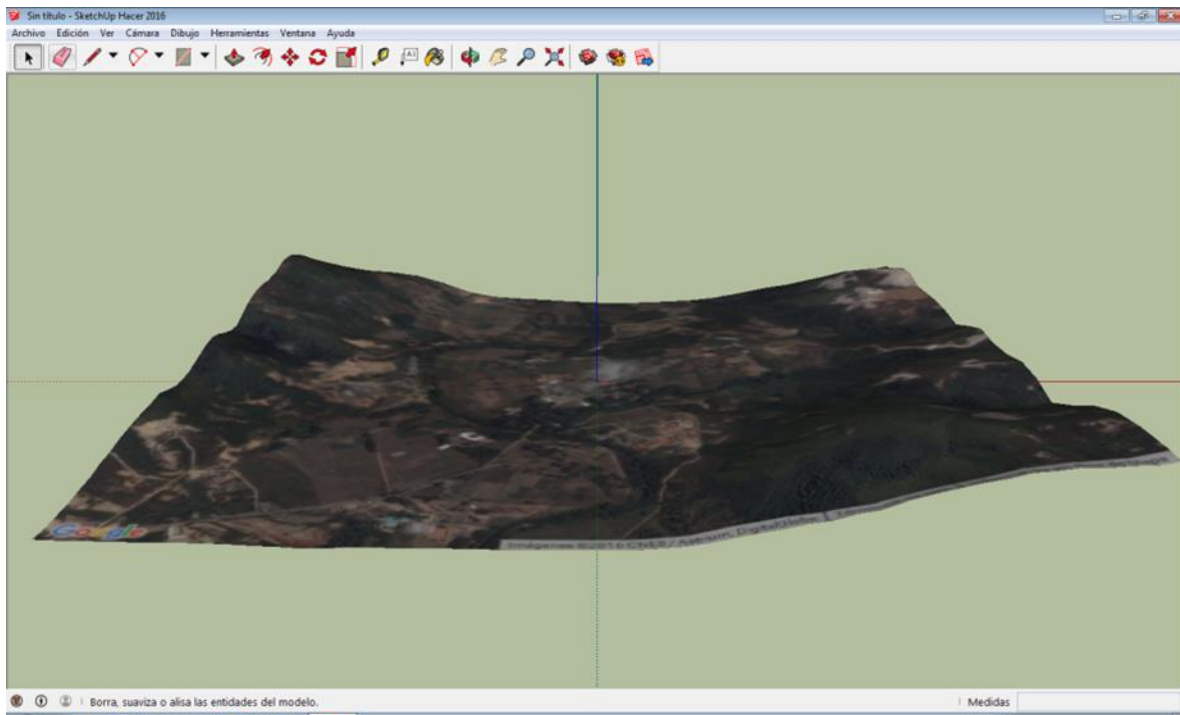


Figura 32. Geografía del terreno alrededor de la UFPSO.
Fuente: Autores del proyecto

El siguiente paso en la creación del terreno fue exportar el plano, para esto se usó el comando en el menú Archivo/Exportar/Modelo 3D, que muestra la interfaz donde se le asigna el nombre y la extensión que tendrá el archivo. Se empleó la extensión obj, que corresponde a un formato abierto para representación de geometrías.

Ese archivo se importó en Blender, a través del menú “Archivo/Importar/Wavefront (.obj)”, obteniendo así un modelo 3D del terreno que podía ser modificado y adaptado de acuerdo a las necesidades. Se comenzó por limpiar la Maya eliminando posibles vértices repetidos, verificando que fuera un solo objeto y que la textura estuviera correctamente asignada, esto con el fin de evitar errores en los procesos posteriores en la fase de creación del terreno. Como resultado se obtiene una maya limpia del terreno como se puede apreciar en la Figura 33.

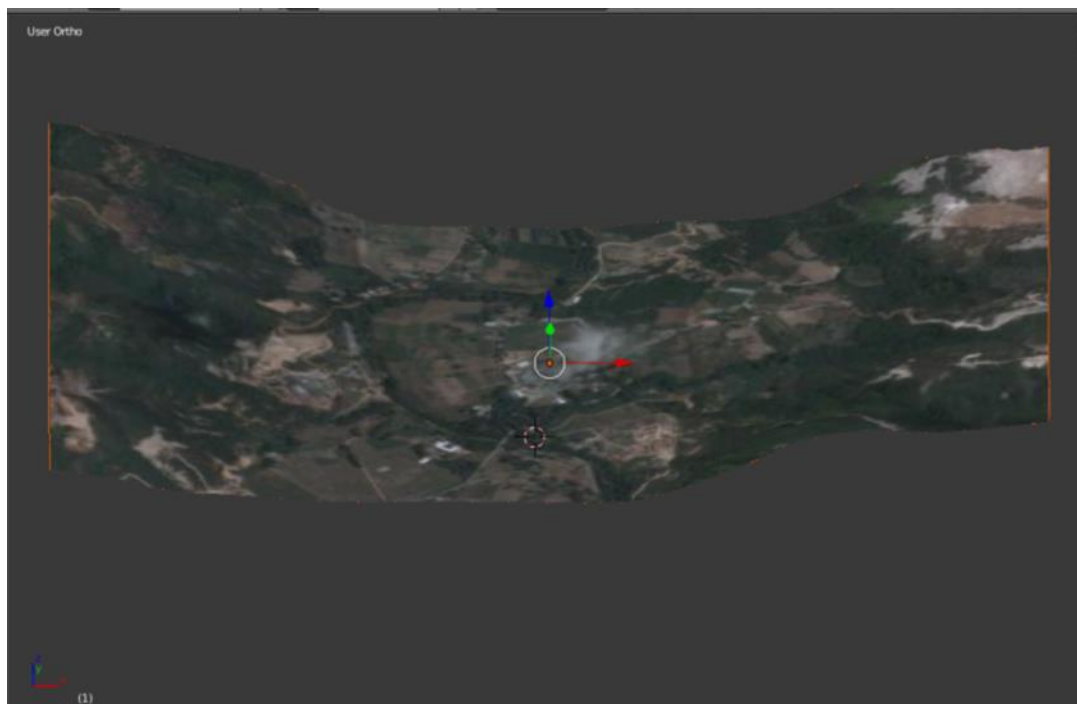


Figura 33. Maya del terreno importada en Blender.

Fuente: Autores del proyecto.

Paralelamente a este proceso, se llevó a cabo la creación de unos modelos de cubo, en Blender, sobre el plano general de la UFPSO que se había generado inicialmente, estos representaban algunos edificios y varias zonas de la Universidad, ver Figura 34, el objetivo de estos fue facilitar la localización y posterior ajuste de los dos planos, plano de la universidad y

plano de alturas, así como también proveer puntos de referencia para la siguiente etapa en el proceso de desarrollo del proyecto.

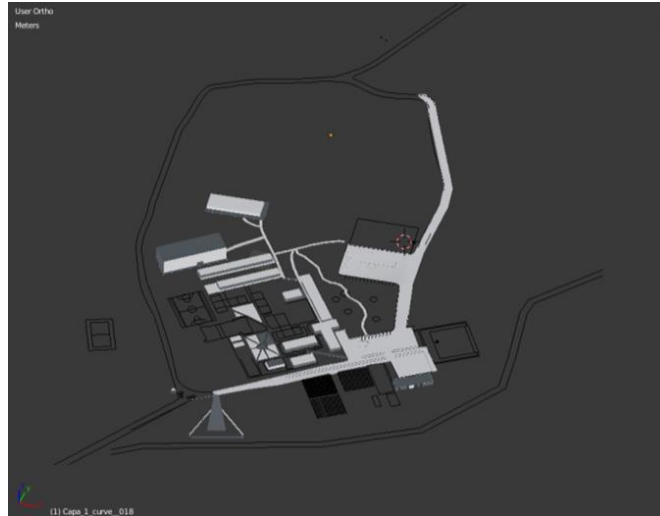


Figura 34. Representación por cubos de algunas zonas de la Universidad.
Fuente: Autores del proyecto.

Una de las últimas etapas realizadas en Blender para la creación del terreno y que constituye un paso muy importante, consistió en la alineación de los dos modelos, ver Figura 35. En este paso se importó el modelo del plano de la Universidad dentro del modelo de alturas, para esto desde el modelo de alturas se ejecutó el comando “Archivo/Append”, que permite importar contenido desde otros archivos de Blender, cuando ya se tenían los dos modelos en un archivo se llevó a cabo la superposición de los dos modelos posterior alineación.



Figura 35. Alineación de los dos modelos.
Fuente: Autores del proyecto.

Luego de hacer coincidir ambos modelos, se guardaron en archivos separados. Cabe aclarar que ya que estos compartían el mismo punto como centro, al volver a juntarlos nuevamente ya estarían alineados, esto es importante para el trabajo que se llevaría a cabo en Unity más adelante.

En el siguiente paso se creó un proyecto en Unity, al cual se importaron el modelo 3D del terreno y el modelo 3D del plano de la Universidad. La importación de estos asset, que es como se le llama dentro de Unity a cualquier elemento que pueda ser utilizado dentro del proyecto (Unity3D, 2015), se llevó a cabo presionando botón derecho del mouse sobre el panel Project y seleccionando la opción “Import New Asset”, ver Figura 36.

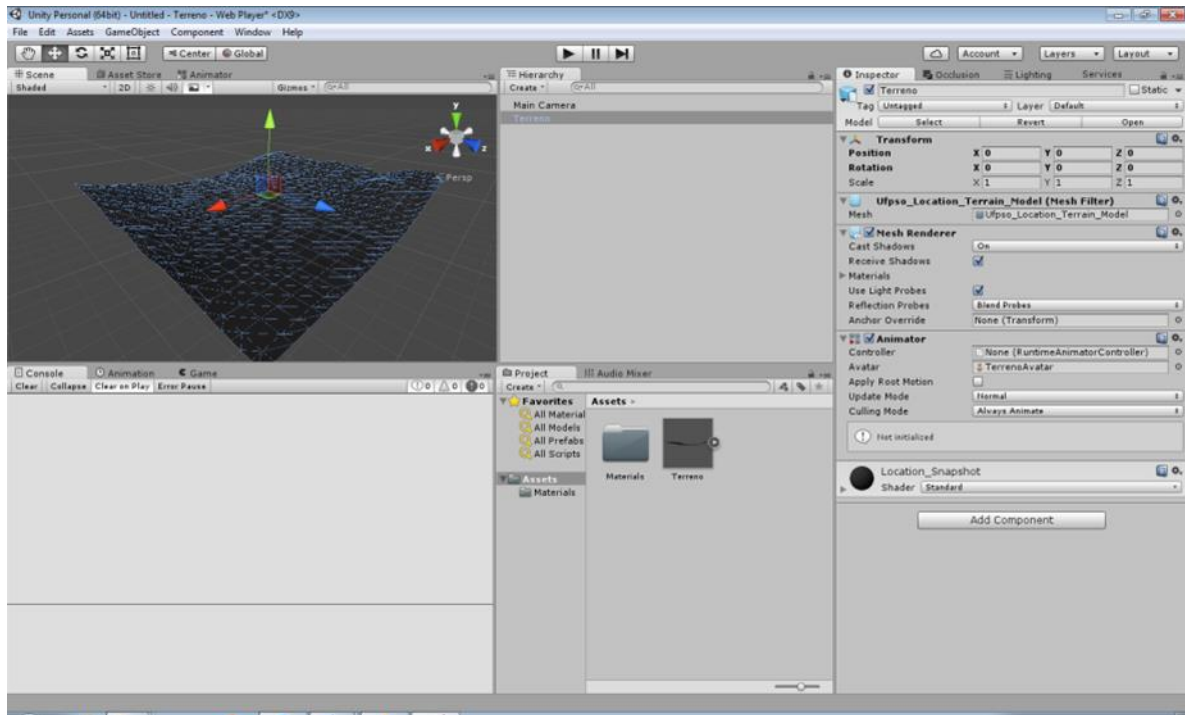


Figura 36. Importación del asset del terreno en Unity.
Fuente: Autores del proyecto.

Hasta este punto se tiene un modelo 3D básico de la geografía del terreno. Unity cuenta con un GameObject llamado Terrain que permite generar terrenos complejos siendo estos gestionados en forma automática por Unity, obteniendo así optimizaciones de rendimiento en tiempo de ejecución, así como herramientas en modo editor que permiten una cómoda configuración del terreno.

La conversión del modelo Inicial del terreno en un GameObject de tipo Terrain, se llevó a cabo a través del script Object2terrain (Haines, 2016), escrito en C Sharp, ver Apéndice C. **Script Object2Terrain** Este toma la maya del asset que se importó y genera un GameObject Terrain que tiene las mismas características de altura y dimensiones que el terreno original. En la Figura

37 se observa el resultado luego de emplear dicho script. En la parte izquierda se aprecia en color rosado, resaltado así para facilitar su identificación, el modelo que se importó inicialmente desde Blender y con base en el cual se generó el otro terreno que se ve en la parte derecha de dicha figura.

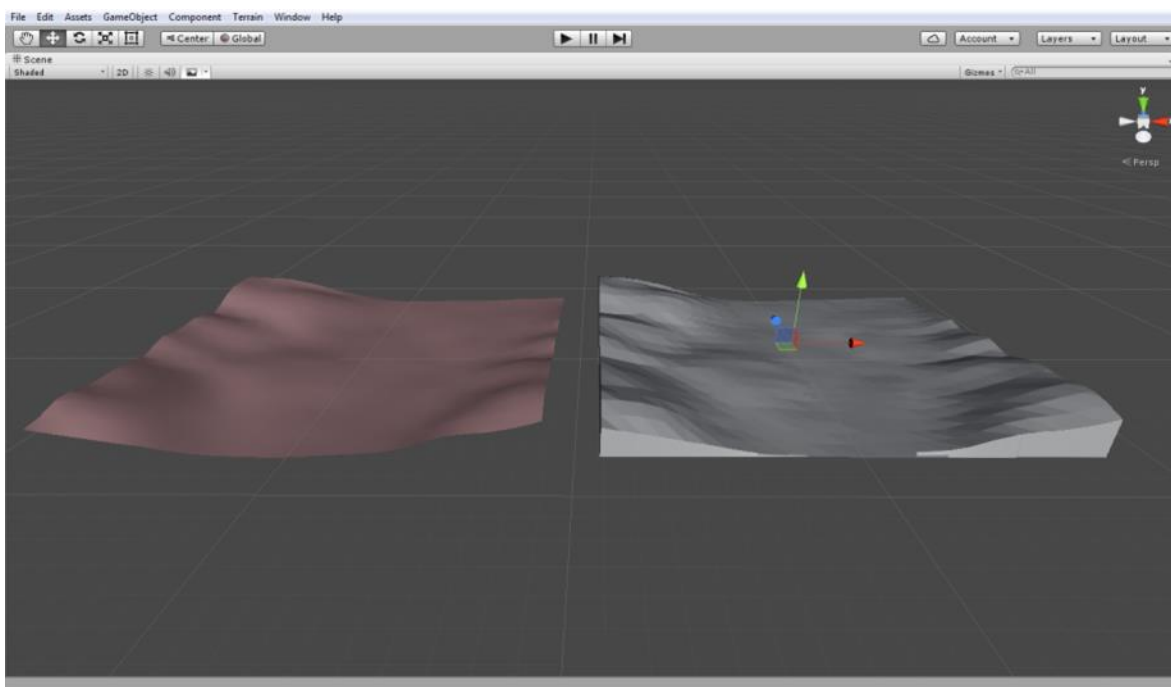


Figura 37. Terrenos dentro de Unity, izquierda Maya Inicial del terreno y derecha GameObject generado con el script.

Fuente: Autores del proyecto.

A continuación se procedió a alinear los dos terrenos para hacerlos coincidir en posición y altura, para lo cual se ajustó la posición del GameObject Terrain moviéndolo hasta la posición del asset terreno, ver Figura 38. Una vez conseguido esto se ocultó el asset terreno dejando solo el nuevo Terrain que será la base para el resto de la integración de los diferentes componentes del recorrido virtual.

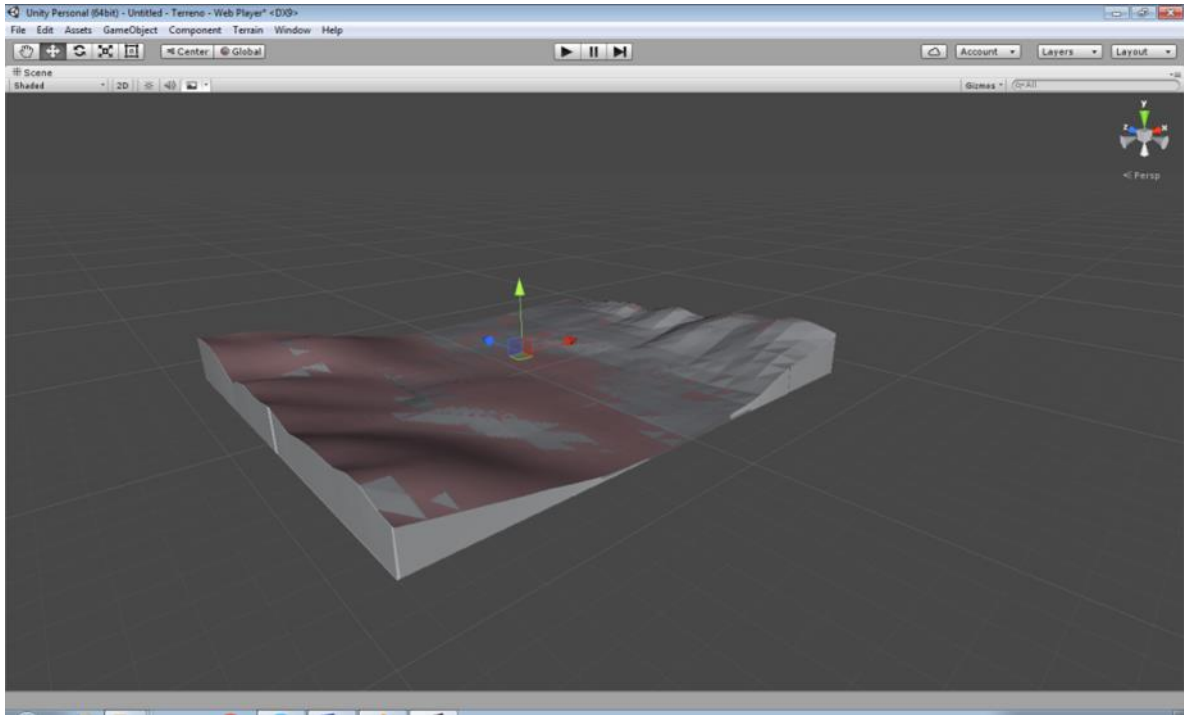


Figura 38. Alineación de los terrenos dentro de Unity.
Fuente: Autores del proyecto.

Una vez alineados los terrenos se procedió con la importación del asset del plano de la universidad, de tal forma que al incluirlo en la escena con algunos arreglos mínimos este quedo ubicado en la parte que le correspondía ver Figura 39 , esto gracias al ajuste que se realizó dentro de Blender.

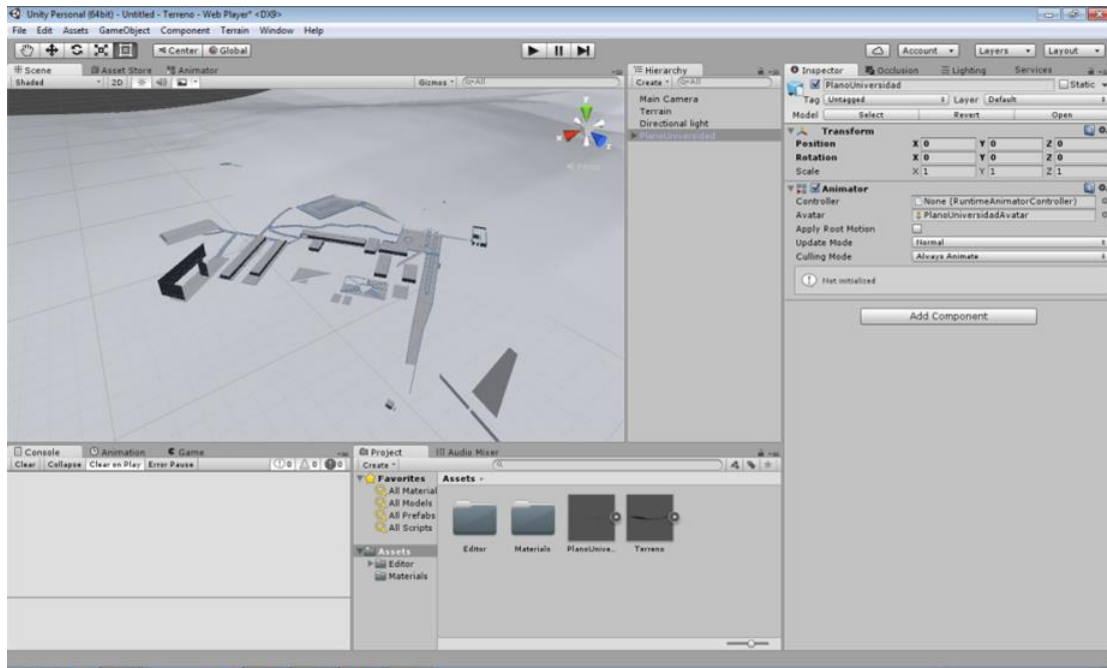


Figura 39. Integración del GameObject Terrain y el asset del plano de la Universidad.
Fuente: Autores del proyecto.

Con el modelo 3D del plano de la Universidad cargado como punto de referencia, el siguiente paso fue la configuración del terreno para darle un aspecto cercano al real para esto se aprovecharon las herramientas con las que cuenta Unity para trabajar con GameObject del tipo Terrain ver Figura 40.

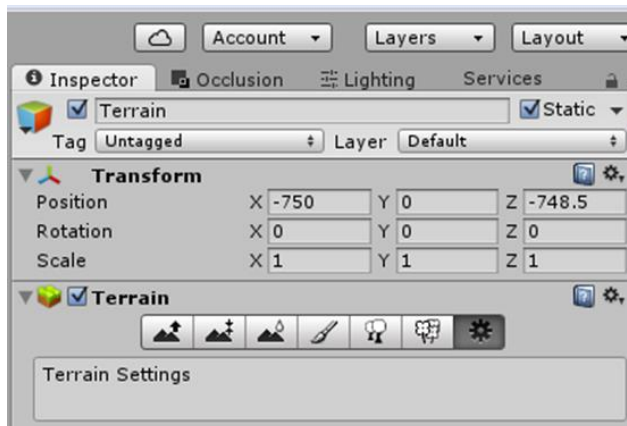


Figura 40. Herramientas para trabajar con terrenos ofrecidas por Unity en el Panel del Inspector.
Fuente: Autores del proyecto.

Las pestañas de derecha a izquierda son Terrain Settings, Paint Details, Place Trees, Paint Texture, Smooth Height, Paint Height, Raise/Lower Terrain; Terrain Settings contiene toda la información del terreno desde la configuración básica del terreno, pasando por cómo se comportan los árboles que se ubiquen y las características del viento. Así mismo Paint Details permite pintar texturas de pasto y detalles sobre el terreno, Place Trees es una función de Unity cargar asset de árboles prefabricados y ubicarlos sobre el terreno, Pain Texture es la herramienta con la cual se pinta la textura sobre el terreno y Smooth Height, Paint Height y Raise/Lower Terrain son herramientas para cambiar la geografía del terreno a través de la modificación de las alturas.

El terreno que se obtuvo fue ajustado a través de la utilización de estas herramientas llegando al resultado mostrado en la Figura 41, en la que se aprecia el gran tamaño del terreno en relación a la zona central que corresponde al terreno cercano a la UFPSO.

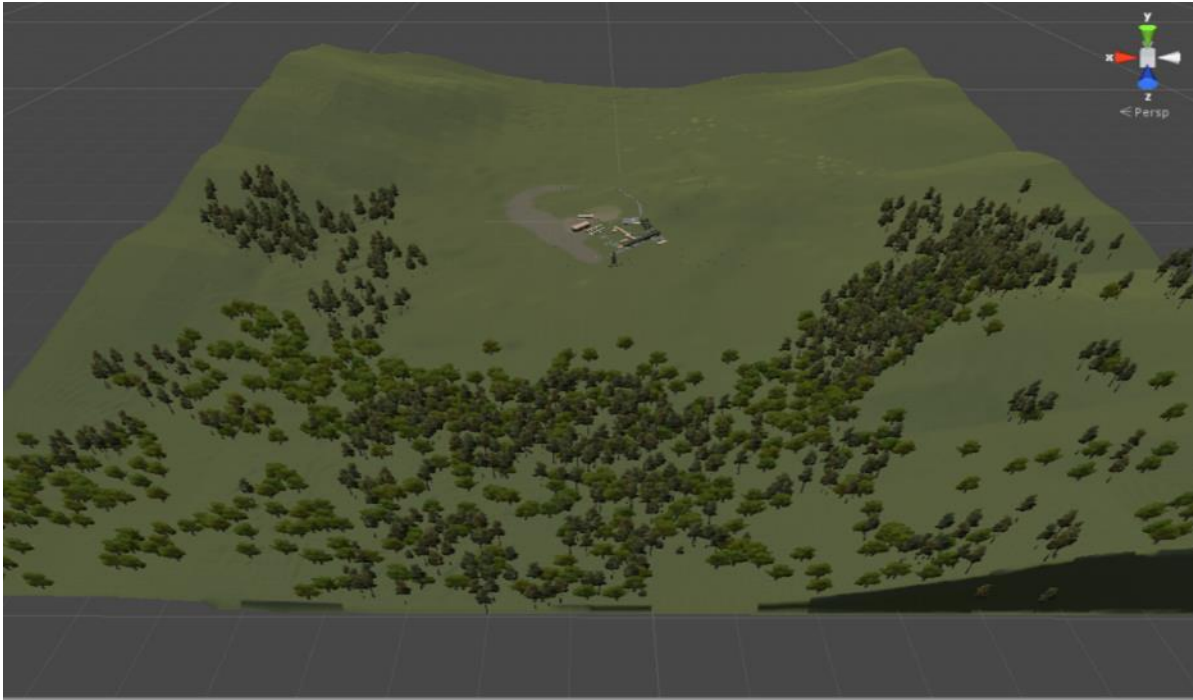


Figura 41. Terreno obtenido luego de aplicar texturas y ubicar árboles.
Fuente: Autores del proyecto.

Este terreno tenía un ancho de 2000 unidades por 1996 unidades de largo, la superficie estaba cubierta por texturas de materiales rocosos y una gran cantidad de vegetación y árboles, dándole así una apariencia muy real, con lo que se consiguió lo que se buscaba en un principio. Pero al ser tan grande, los costes para renderizar la escena en tiempo real eran bastantes altos a pesar de las optimizaciones automáticas de terreno realizadas por Unity. Dado que se necesitaba que fuera accesible por computadores con hardware de características estándar, se decidió realizar una reducción del terreno hasta un 25% del tamaño inicial pero ya que se querían conservar los detalles de los entornos se empleó también otra herramienta para crear un skybox que tuviera una imagen del entorno.

El primer paso fue la construcción del skybox, para esto se empleó el asset 360 Panorama Capture disponible en la AssetStore de Unity (Unity3D, 2015) de forma gratuita, este asset permitió obtener las seis imágenes para armar el cubo del skybox ver Figura 42.

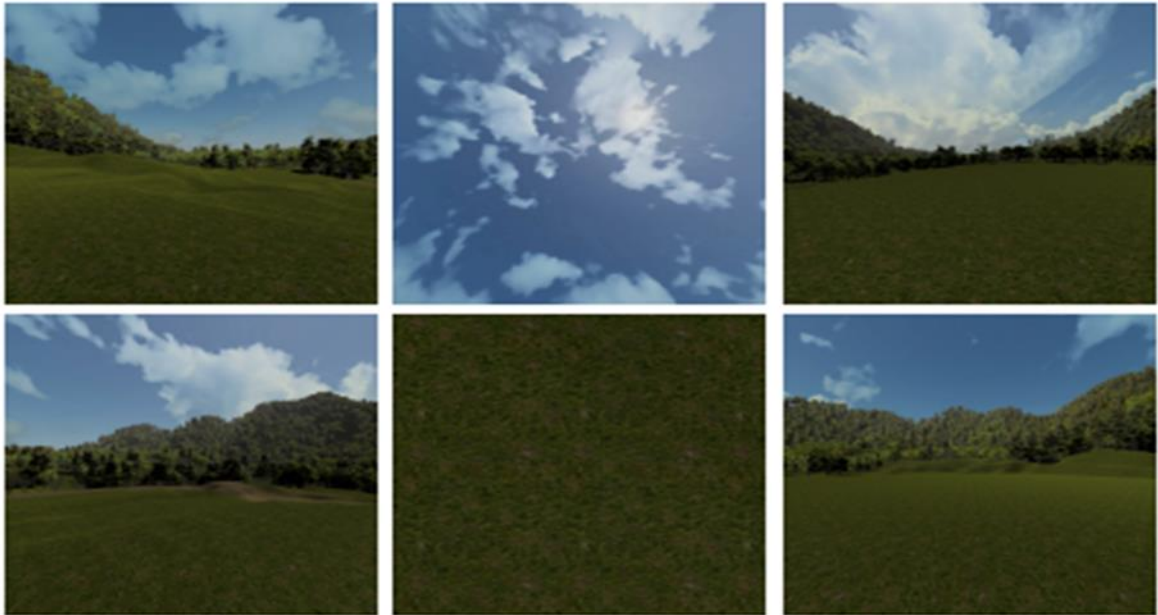


Figura 42. Imágenes 360 grados para skybox.
Fuente: Autores del proyecto.

Teniendo estas imágenes se procede a crear un nuevo material dentro de Unity y selecciona el shader Skybox/ 6 Sided, de esta forma ya se tiene el skybox; posteriormente se recorta el terreno, dejando solo el 25% del tamaño inicial llegando así a tener 500 unidades de ancho por 499 unidades de largo, para este proceso se empleó el asset SplitTerrain (Dvornik, 2013).

El script trabaja dividiendo el terreno que se le pasa como argumento en cuatro partes, por lo que fue necesario hacer divisiones sucesivas del terreno hasta alcanzar el tamaño deseado,

como se observa en la Figura 43, el primer corte, líneas amarillas, dividió el terreno inicial en cuatro terrenos; posteriormente cada terreno fue dividido nuevamente, las líneas rojas muestran por donde se cortó cada terreno. Al final del proceso se conservaron los cuatro terrenos centrales que corresponden a la zona donde está ubicada la UFPSO, el resultado final se puede apreciar en la Figura 44.

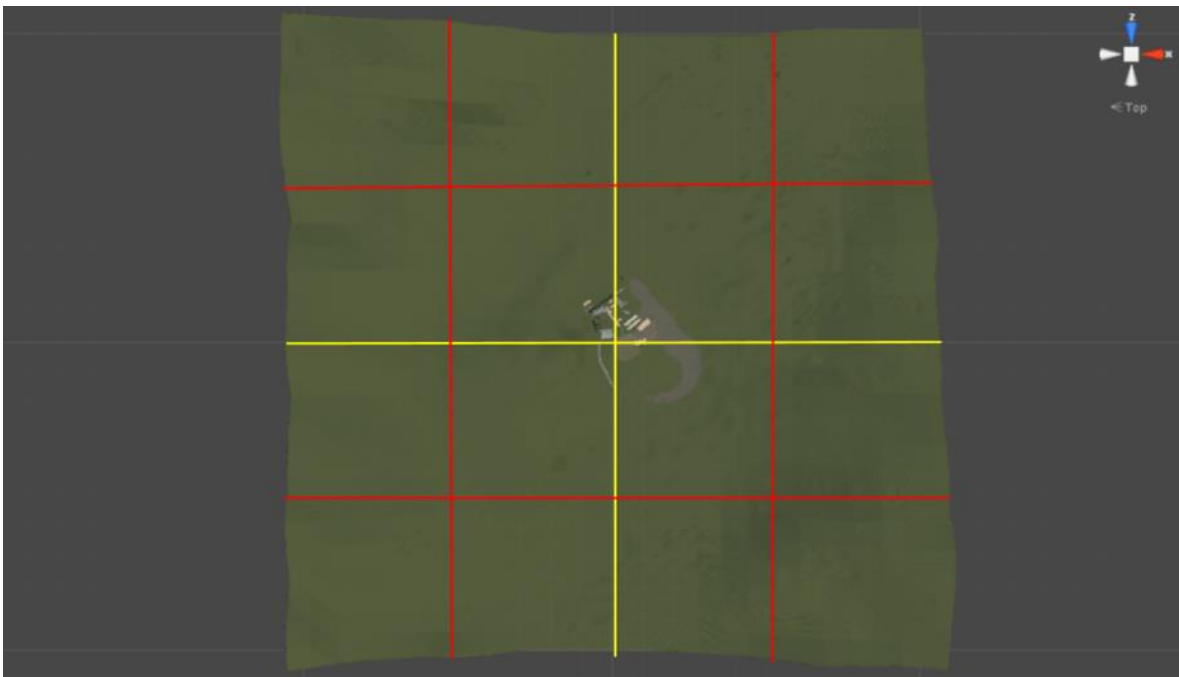


Figura 43. Cortes realizados al terreno inicial. .
Fuente: Autores del proyecto.

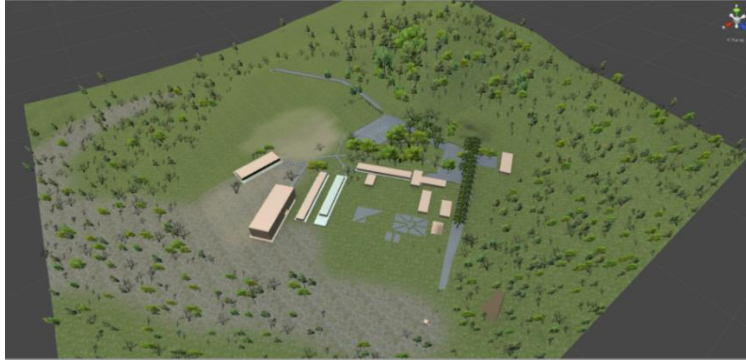


Figura 44. Terreno final.
Fuente: Autores del proyecto.

Los cuatro terrenos que en conjunto forman el terreno del recorrido virtual de la UFPSO están perfectamente alineados entre sí, sin embargo al mirar de cerca se nota cierta diferencia entre las texturas para solucionar esto se creó el script UnirTerreno, ver Apéndice D, la función de este script es relacionar entre si los cuatro terrenos y que sea visto como uno solo para evitar diferencias en las texturas. Para esto debe adjuntarse a cada terreno una copia de dicho script y pasarle como argumentos los terrenos que lo rodean.

Finalmente se agregó el skybox que se había creado anteriormente, ver Figura 45, esto ocasiona que cuando se mira desde el terreno se tenga la sensación de un terreno sin fin como se puede observar en la Figura 46.



Figura 45. Vista superior del terreno con el skybox.
Fuente: Autores del proyecto.



Figura 46. Vista desde la superficie del terreno con el skybox.
Fuente: Autores del proyecto.

1.17.2 Importación de los modelos 3D a Unity. En esta parte del proyecto se importaron los objetos desarrollados en la fase de modelado y se realizó su ubicación en la posición correspondiente, para esto primero se situaron los modelos de Casona y Anexos Académicos , posteriormente se introdujeron los objetos que conforman cada una de las oficinas basándose en imágenes tomadas en los diferentes lugares ver Figura 47, para finalizar se agregan árboles y vegetación con el fin de dar una mejor vista a la escena, ver Figura 48.

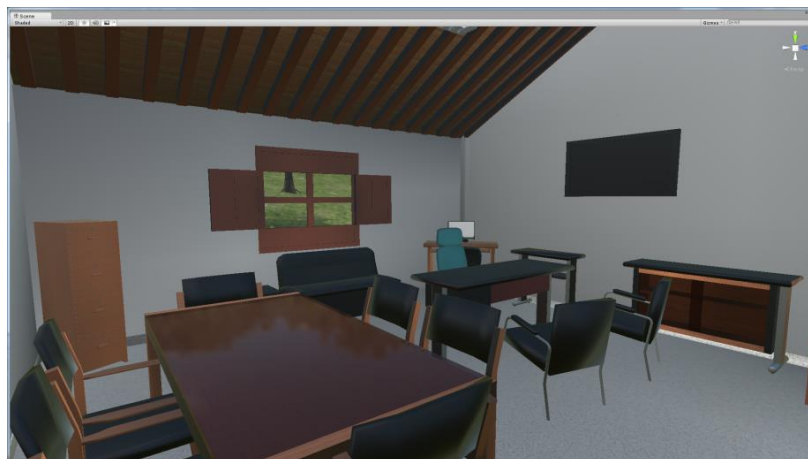


Figura 47. Vista virtual de una oficina
Fuente: Autores del proyecto.



Figura 48. Vista exterior del campus
Fuente: Autores del proyecto.

1.17.3 Creación del Mini-Mapa

Con el fin de mejorar la navegabilidad dentro del recorrido y brindar facilidad de ubicación para el usuario, se creó un mini-mapa que es el encargado de mostrar la posición del usuario dentro del recorrido virtual. Para esto se utilizó una imagen de todo el terreno desde una vista superior, ver Figura 49, posteriormente esta imagen es utilizada por el script miniMapa para generar una representación del terreno con un punto central que simboliza la ubicación del usuario, ver Apéndice I.



Figura 49. Vista Superior del Terreno
Fuente: Autores del proyecto.

También es posible ampliar la ventana del mini-mapa para ver en forma más detallada la información de este, para esto el usuario debe presionar la tecla M y el script miniMapa generara una versión más grande del mini-mapa, dicha ventana tendrá el 80% del alto de la ventana en la cual se observa la aplicación, lo que quiere decir que el tamaño del mapa que vea el usuario dependerá de la resolución que este esté usando, ver Figura 50 y Figura 51.

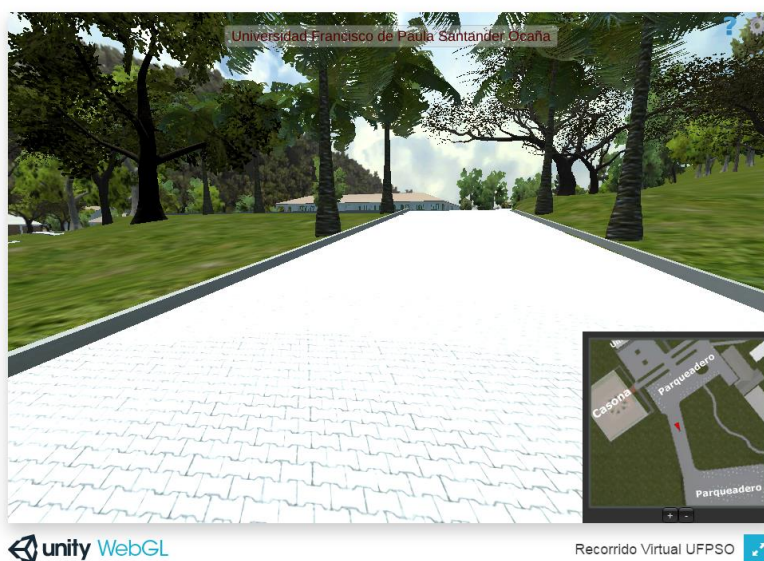


Figura 50. Vista normal del Mini-mapa
Fuente: Autores del proyecto.



Figura 51. Mini-mapa maximizado al presionar la tecla P
Fuente: Autores del proyecto.

1.17.4 Optimizaciones. Debido a problemas de rendimiento en cuanto a requisitos para una óptima ejecución, mal funcionamiento de algunas funciones y métodos de optimización necesarios, encontrados durante pruebas realizadas previamente se tomaron ciertas medidas que se enumeran a continuación.

1.17.4.1 Occlusion Culling. El Occlusion Culling es una función que permite que los objetos que no se muestran en pantalla o estén detrás de otros objetos no sean renderizados por la cámara. Debido a que esta función no trabaja correctamente en la versión gratuita del Unity se crearon unos scripts, ver Apéndice F, que harán aparecer o desaparecer los objetos más complejos, esto se controla mediante triggers ubicados en diferentes zonas de los edificios, ver Figura 52, que son activados o desactivados dependiendo de la posición de la cámara dentro del recorrido, logrando así reducir la cantidad de vértices que se están calculando en la escena y optimizar drásticamente el rendimiento y el consumo de recursos por parte de la aplicación.

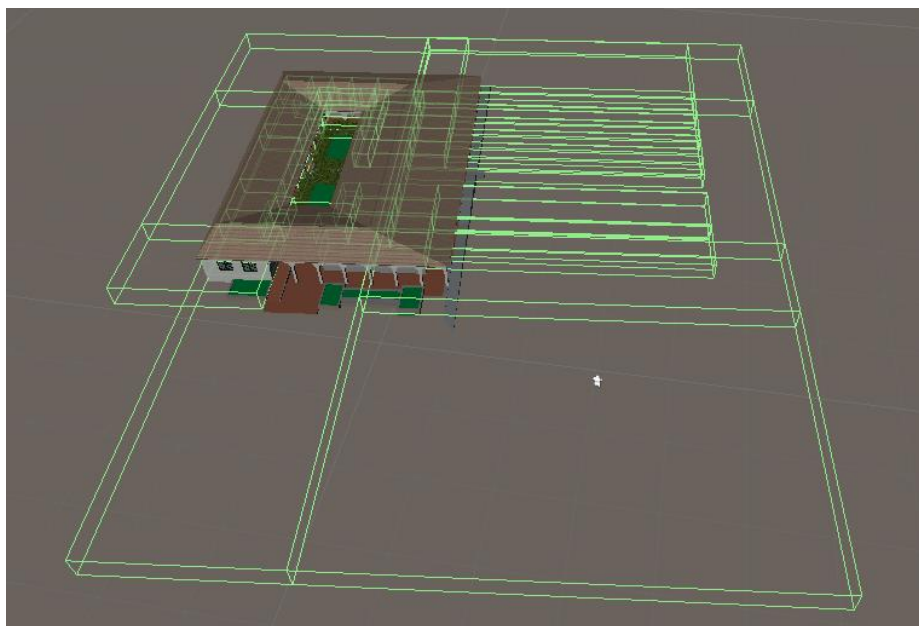


Figura 52. Ubicación de los triggers en el edificio de Anexos Académicos
Fuente: Autores del proyecto.



Figura 53. Script desactivado. 3.2 M de vértices en la escena
Fuente: Autores del proyecto.

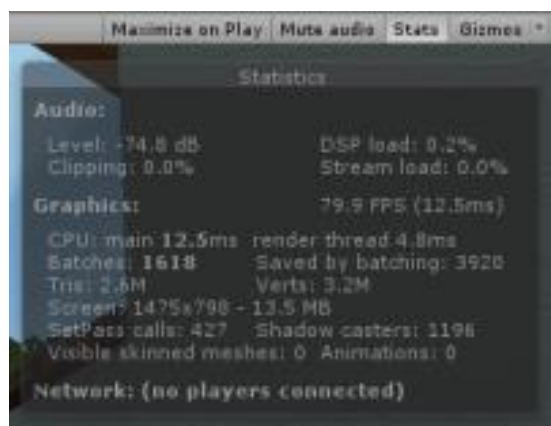


Figura 54. Estadísticas de rendimiento de la escena con el script desactivado
Fuente: Autores del proyecto.

En la Figura 53 se muestra una escena sin el script activo dando una cantidad de 3.2 millones de vértices y 2.4 millones de triángulos calculados en la vista, ver Figura 55, lo que se traduce en una gran exigencia de recursos computacionales, lo cual puede generar una baja fluidez, ocasionando un comportamiento incómodo para el usuario. Por otra parte tomando la

misma posición en la escena, ver Figura 55, y al activar el script se obtienen 816 mil vértices y 690 mil triángulos llegando a consumir aproximadamente el 25% de los recursos necesarios inicialmente, como se muestra en la Figura 56.



Figura 55. Script activado. 690 K vértices en la escena.
Fuente: Autores del proyecto.



Figura 56. Estadísticas de rendimiento de la escena con el script activado.
Fuente: Autores del proyecto.

1.17.4.2 Iluminación y Ambient Occlusion. Se tuvieron en cuenta diferentes aspectos en cuanto al rendimiento de la aplicación, como lo son la iluminación pre-calculada, ya que la iluminación en tiempo real puede ser una tarea bastante exigente. Realizar de antemano los mapas de iluminación reduce en gran medida la cantidad de recursos computacionales consumidos por la aplicación. Teniendo en cuenta lo anterior y con el fin de hacer que el recorrido virtual sea lo más accesible posible para los usuarios finales, se realizó un precálculo de los niveles de iluminación en las diferentes escenas que componen el recorrido, teniendo en cuenta una iluminación global precomputarizada de diferentes fuentes, como el cielo y una luz direccional, que hará un efecto similar a la iluminación producida por el sol, ver Figura 57.

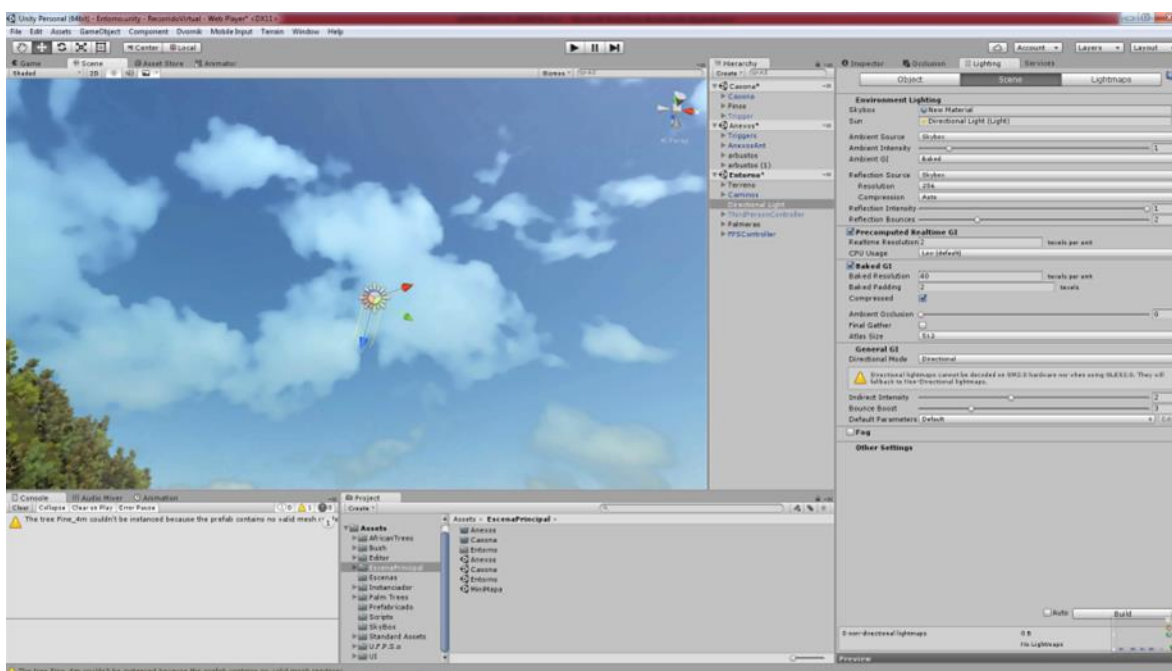


Figura 57. Parámetros de Configuración de Iluminación en Unity.
Fuente: Autores del proyecto.

Además de los parámetros y fuentes de iluminación global, se configuro un material con un parámetro de emisión y este se agregó a todas las lámparas presentes en las oficinas, de tal forma

que al generar los mapas de luces toda esta información es precalculada y guardada, por lo tanto momento de correr la aplicación se verá como estas lámparas emiten luz en las oficinas dando una apariencia más realista, con un bajo consumo de recursos en comparación a si fuera iluminación calculada en tiempo real, ver Figura 58. En caso de no usarse una iluminación la calidad de la vista se reduciría drásticamente al no permitir detallar las sombras y los diferentes tonos de iluminación.

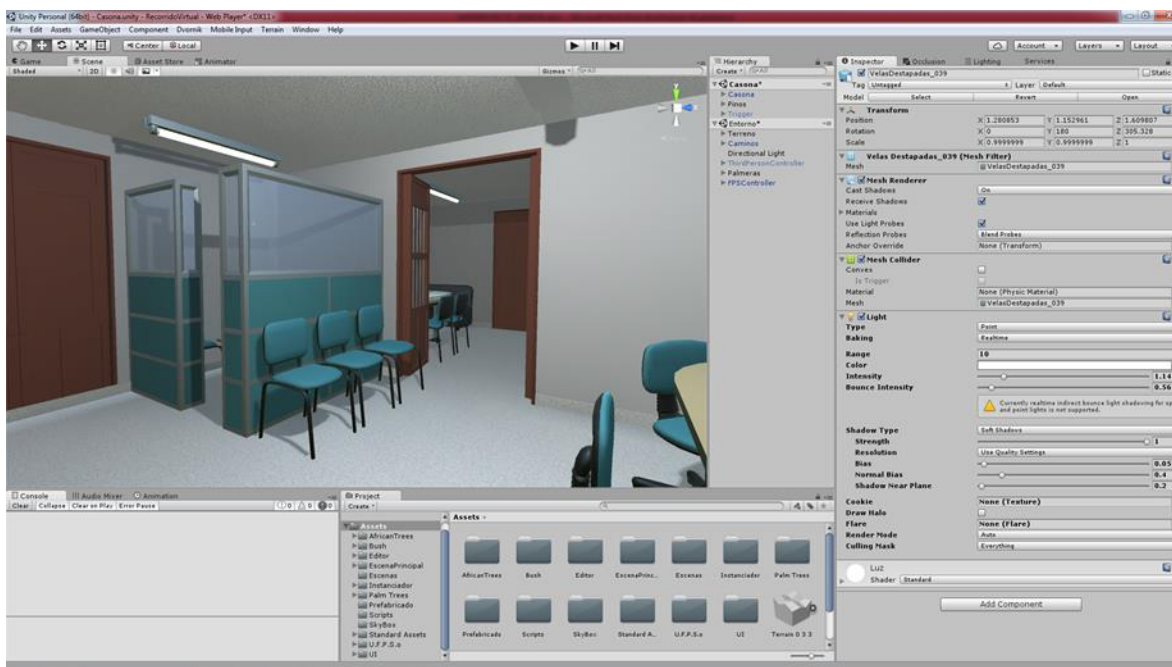


Figura 58. Vista de una oficina con iluminación. .
Fuente: Autores del proyecto.

1.17.5 Creación de la escena del dialogo de Santanderito. Para la realización del dialogo, se procedió a crear un libreto, ver Apéndice G, en el que se habla muy brevemente sobre la historia de la Universidad Francisco de Paula Santander Ocaña, las zonas o lugares a las que se podrá acceder durante el recorrido, además de la bienvenida y una sencilla explicación de la

mecánica del recorrido. Posteriormente y debido a la carencia de una voz oficial para el personaje, se utilizaron voces generadas por computador, mediante un text to speech (TTS en adelante) con las cuales se crearon los audios del dialogo.

Posteriormente con el uso del programa Papagayo lip-sync se ajustaron los fonemas que utiliza el personaje para cada dialogo establecido previamente, en base al fotograma correspondiente a los tiempos del audio, Figura 59; con el documento generado por papagayo, se realizó un script que permitió generar la animación de los movimientos vocales en Unity, ver Apéndice H, en base a los fonemas establecidos en el personaje y se usó una base de movimientos corporales en estado de relajación, sobre los cuales se combinaron los audios y movimientos faciales, logando una impresión natural en la animación del personaje, Figura 60.



Figura 59. Sincronización de fonemas con Papagayo
Fuente: Autores del proyecto.

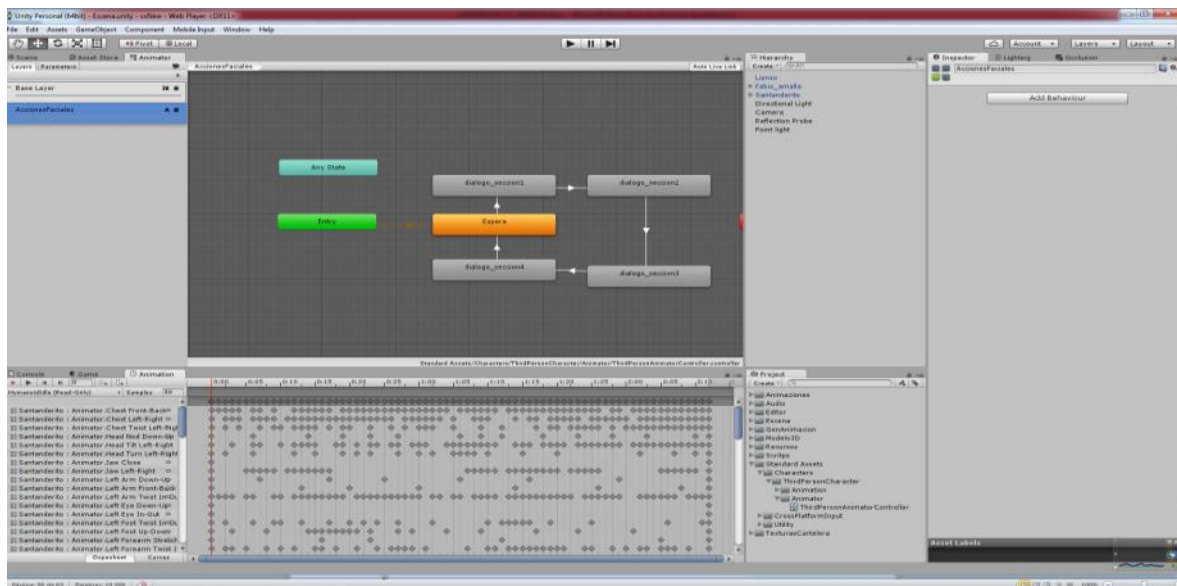


Figura 60. Animación en Unity
Fuente: Autores del proyecto.

Para completar la escena se ambientó la animación dentro del auditorio Fabio Amalla, en la cual se realiza el discurso por parte del personaje en forma de presentación al público, Figura 61

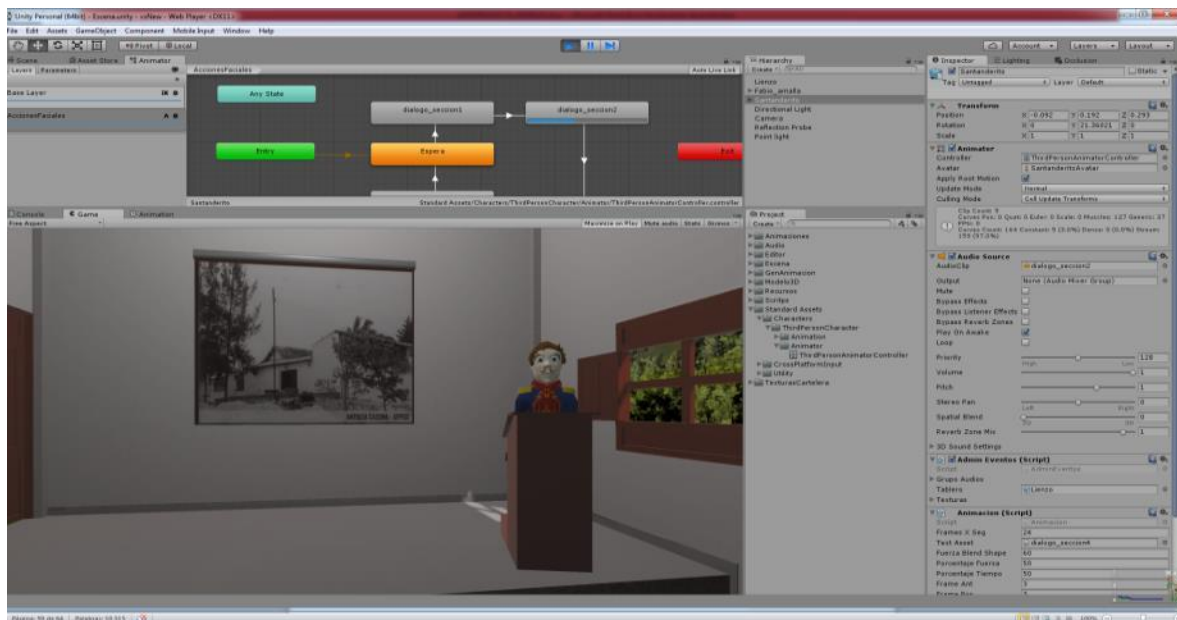


Figura 61. Escena del dialogo introductorio.
Fuente: Autores del proyecto.

1.17.6 Creación de la Escena de Inicio y la Escena del Menú de Inicio. Hasta el momento el recorrido cuenta con dos escenas la escena del recorrido del campus, ver Figura 62, y la escena del dialogo de Santanderito, ver Figura 63, en esta fase del desarrollo del proyecto se describe el proceso de creación de la escena de inicio y la escena del menú de inicio.



Figura 62. Escena del recorrido del Campus.
Fuente: Autores del proyecto.



Figura 63. Escena dialogo de Santanderito en el auditorio Fabio Amaya.
Fuente: Autores del proyecto.

1.17.6.1 Escena de Inicio. Esta escena es la primera que se carga al abrir la aplicación, en ella se presenta el escudo de la Universidad que aparece por un lapso de 5 segundos, ver *Figura 64*, y luego da paso a la escena del menú de inicio.



Figura 64. Escena de Inicio de la aplicación.
Fuente: Autores del proyecto.

1.17.6.2 Escena del Menú de Inicio. Con el fin de dar una mejor presentación a la aplicación se agregó un menú al inicio, ver *Figura 65*, esta escena es la segunda en cargar y en ella se muestra un menú con cuatro botones, el primero que corresponde a la opción Iniciar este hace cargar la escena del dialogo de Santanderito y seria el que se encarga de iniciar el recorrido, además están los botón de ayuda, configuración y créditos, al accionar cada uno de estos se despliega al usuario el panel correspondiente de acuerdo a la opción escogida.



Figura 65. Escena del menú de inicio.
Fuente: Autores del proyecto.

Al presionar el botón de la ayuda se muestra un panel en el que aparece información sobre los controles del recorrido, esto con el fin de que el usuario aprenda cual es la mecánica para desplazarse dentro de este, ver Figura 66.



Figura 66. Panel de ayuda del menú de inicio.
Fuente: Autores del proyecto.

El botón configuración es el encargado de activar el panel de configuración, en el cual se le permite al usuario fijar algunos parámetros de ejecución tales como la calidad de la imagen, el volumen general, las voces, la música de fondo y los sonidos del ambiente, ver Figura 67.

El botón de créditos despliega un panel en el que se muestra información de todas aquellas personas que estuvieron involucradas en el desarrollo de la aplicación, así como las principales herramientas que se utilizaron en el desarrollo de esta, ver Figura 68.



Figura 67. Panel de configuración del menú de inicio.
Fuente: Autores del proyecto.



Figura 68. Panel de créditos del menú de inicio.
Fuente: Autores del proyecto.

1.18 Incorporar la Aplicación Desarrollada del Recorrido Virtual en un Sitio Web, Facilitando el Acceso por Medio de Internet.

1.18.1 Parámetros de Compilación. En el momento de la compilación se tuvieron en cuenta varios aspectos; ya que la aplicación va dirigida a personas de diferentes lugares la aplicación se construyó para navegadores web, en los cuales se permiten dos opciones, el uso del plugin de Unity, el cual es una versión estable pero que próximamente dejara de tener soporte y la compilación para WebGL, la cual aún está en una fase de desarrollo.

En la versión del WebPlayer que da uso al plugin de Unity, se utilizó el asset llamado AssetBundle, permitiendo a los usuarios descargar y utilizar la aplicación de una manera más fluida, ya que este habilita el uso de la aplicación sin que esta se haya descargado completamente, mostrando una imagen de carga mientras se adquiere el contenido faltante para la escena en cuestión, ver Figura 69.

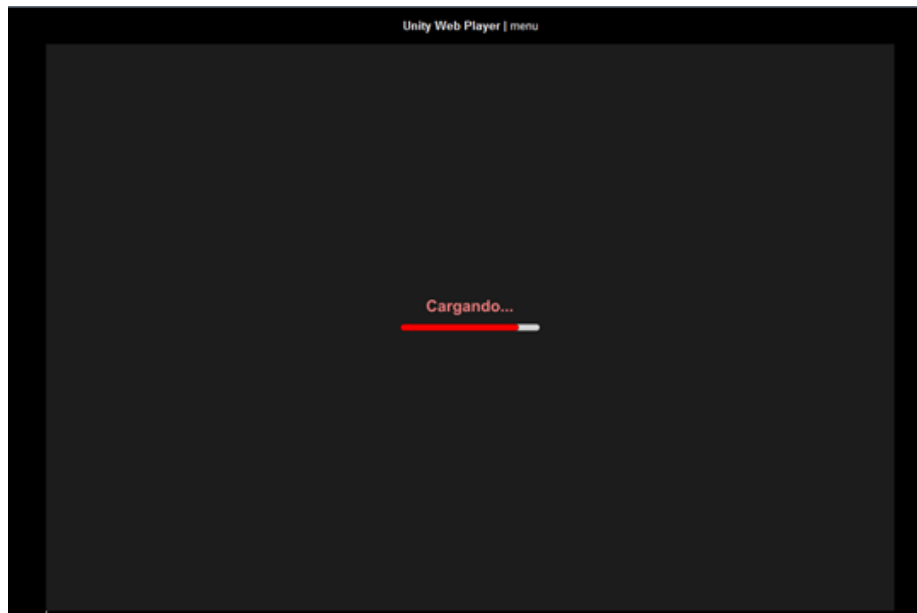


Figura 69. Fase de carga del WebPlayer.
Fuente: Autores del proyecto.

La versión WebGL del aplicativo necesita descargar completamente su contenido, antes de que el usuario final pueda acceder a este, ver Figura 70.

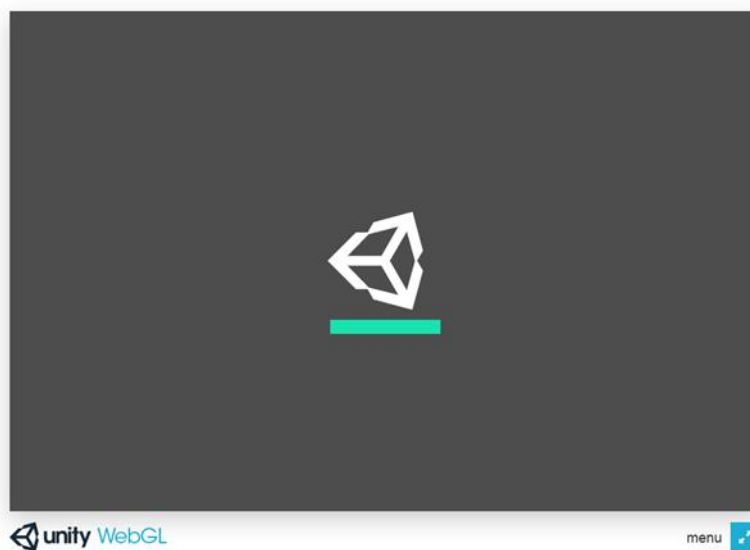


Figura 70. Fase de carga del WebGL.
Fuente: Autores del proyecto.

Ambas versiones se configuraron para trabajar con una resolución inicial de 960px de ancho por 640px de alto; por otro lado el WebGL trabaja con una asignación de memoria estática, por lo tanto le se asignó 512MB con los que deberá correr la aplicación; a diferencia del WebGL, el WebPlayer trabaja con una asignación de memoria dinámica, por lo tanto no fue necesario darle un valor.

Debido a la decisión tomada por Unity de discontinuar la opción de compilación para la versión WebPlayer en futuras versiones del software; debido a esto se decidió realizar la compilación del presente proyecto en la versión WebGL, esto teniendo en cuenta las mejoras de rendimiento, anunciadas por Unity.

Para un buen rendimiento de la aplicación, se recomienda al usuario el uso de los parámetros de calidad descrita en la **Tabla 13**, en relación a la configuración del pc del cual este haciendo uso.

Tabla 13.
Configuración de calidad con respecto al rendimiento del computador.

Configuración de Calidad	CPU	Memoria RAM	Video o GPU
Muy Baja	Intel Core 2 Duo o equivalente	2Gb DDR2	---
Baja	Intel Core 2 Duo o equivalente	2Gb DDR2	Intel HD Graphics 2000 o equivalente
Media	Intel Core I3 o equivalente	4Gb DDR2	AMD Radeon HD 6250 o equivalente
Alta	Intel Core I3 o equivalente	4Gb DDR3	GeForce GTX 460 o equivalente
Muy Alta	Intel Core I5 o equivalente	4Gb DDR3	GeForce GTX 550 o equivalente

Fuente: Autores de Proyecto.

Haciendo uso de la configuración mínima planteada previamente, es posible que el usuario experimente una baja cantidad de imágenes por segundo (FPS por sus siglas en inglés, “Frame Per Second”); llegado el caso de presentarse esta situación, es recomendable reducir la calidad con la que la aplicación se está ejecutando.

1.18.2 Incorporación de la Aplicación del Recorrido Virtual en el Sitio Web. Para dar a conocer el desarrollo del recorrido con mayor facilidad a los usuarios finales, se incorporó la aplicación en un subdominio proveído por la división de sistemas de la Universidad Francisco de Paula Santander Ocaña, ver Figura 71, con las siguientes características:

- Dirección del host: campusvirtual.ufpso.edu.co
- Servidor: Apache, versión 2.4.6, reléase 40.el7.centos, arquitectura x86_64
- PHP, versión 5.4.16, reléase 36.el7_1, arquitectura x86_64.



Figura 71. Recorrido virtual hospedado en el host campusvirtual.ufpso.edu.co.
Fuente: Autores del proyecto.

Conclusiones

En el modelado arquitectónico se encontró cierta dificultad debido a que los modelos en 3D que representan las estructuras existentes, deben tener un gran parecido con la realidad, de tal forma que los usuarios se sientan familiarizados con el entorno, por lo cual fue necesaria la recolección de gran cantidad de información que permitiera ajustar los prototipos a los modelos reales. Dichos modelos no pueden generar una carga excesiva en el motor gráfico, ya que esto afectaría directamente el rendimiento y la experiencia del usuario, por lo tanto fue necesario seleccionar que objetos eran relevantes para el recorrido y cuales prescindibles para el mismo.

En el desarrollo de un recorrido virtual, es importante brindar las herramientas necesarias, tales como, mapas de señalización, rótulos de las oficinas o lugares, así como un mini mapa que permitan al usuario ser consciente de su ubicación dentro del entorno en el cual se está realizando la aplicación, con el fin de permitir al usuario una correlación entre la recorrido y el entorno real.

Para la integración de los modelos 3D en Unity fue importante tener en cuenta el uso de prefabricados, ya que esto permitió reutilizar los elementos ya creados sin un aumento excesivo en el peso, dado que disponer estos objetos en el entorno desde Blender, genero una compilación final poco optimizada en cuanto al uso de objetos y de igual manera producía un mal uso de la iluminación, debido a que al realizar cambios en alguno de estos objetos se reiniciaba

completamente el cálculo de luces para todos los objetos que conformaban el archivo en el cual estaba incluido.

Al momento de integrar las escenas para permitir su uso mediante navegadores web, fue importante tener en cuenta las opciones que brinda Unity, ya que con el WebGL se permite el uso de AssetBundle, pero dicho plugin próximamente dejará de ser soportado, ya que las librerías de la cual este depende, están siendo abandonadas por algunos navegadores; por otro lado el WebGL, es un producto en fase de desarrollo, que mediante el uso de HTML5 permite la simulación de gráficos 3D en un navegador web. Dichas opciones al no ser directamente nativas y estar limitados por el desempeño del navegador, tienen un rendimiento reducido, por lo cual fue necesario tener en cuenta varios tipos de optimización.

Durante el avance de este proyecto, fue interesante experimentar los diferentes roles que se vieron involucrados en la creación del mismo, tales como, diseño, modelado y desarrollo, así como superar los diferentes retos que se presentaron. Aunque no existe una metodología estándar para el desarrollo de este tipo de aplicaciones, da la posibilidad de adaptarse a una técnica conveniente para el equipo de trabajo, haciendo que su progreso sea más versátil.

Recomendaciones

Teniendo en cuenta el alcance de este proyecto, es recomendable continuar con su desarrollo y expansión, a través, de la implementación de nuevas funcionalidades, tales como, acceso a las aulas de clase, edificio Carlos Rizo y biblioteca, de tal forma que los posibles interesados puedan conocer más de nuestra alma mater.

En base a lo anterior y dado que si se desea llevar a cabo futuras expansiones del aplicativo, es recomendable que la universidad adquiriera la licencia educativa de Unity.

Se recomienda que la aplicación sea hospedada en el sitio web de la Universidad Francisco de Paula Santander Ocaña, para dar mayor facilidad a los usuarios interesados en conocer más acerca de la misma; para esto es necesario que la universidad adquiriera una licencia de carácter educativa o comercial del programa Unity.

A medida se avanza en el desarrollo de la aplicación, es posible introducir mejoras en los modelos, por lo que es recomendable almacenar versiones escalables o reutilizables de los prototipos.

En la fase de modelado es importante tener en cuenta que técnica de modelado se va a usar, ya que esto afecta en gran medida los resultados obtenidos del modelado; Para el modelado de

objetos es recomendable el uso de técnicas como box modeling, ya que al iniciar un modelo a partir de una caja o una figura primitiva e ir aumentando su nivel de complejidad hasta lograr la figura deseada, se conseguirá una buena topología de la malla; por otro lado para agregar mayor nivel detalles se puede hacer uso de técnicas como sculpt modeling, aunque esto a costo de una malla muy compleja, la cual no podría ser usada en motores de tiempo real sin una retopología que permita una transición de los detalles del modelo de alta resolución a un modelo menos complejo, pero que mantenga los rasgos más notorios, agregando los menos notorios por medio de texturas generadas del modelo de alta resolución.

Para tener mejor control sobre el proyecto es recomendable adoptar un sistema de control de versiones, lo que permitiría un desarrollo en conjunto por parte de los interesados, además de un respaldo de la información.

Referencias

- Blender. (s.f.). *Introduccion: ¿Qué Es Blender?* Obtenido de <http://wiki.blender.org/index.php/Doc:ES/2.6/Manual/Introduction>
- Brun Amaya, R. (2010). Generación y visualización inmersiva del estadio olímpico de Madrid. (*proyecto de fin de carrera*). Universidad Rey Juan Carlos, Madrid.
- Dvornik, K. (2013). *Ideas: Unity split terrain script*. Obtenido de <http://kostiantyn-dvornik.blogspot.com.co/2013/12/unity-split-terrain-script.html>
- Franco Serrano, V. H. (s.f.). Preparación y exportación de modelos y escenarios para tiempo real. Obtenido de <http://www.ixtli.unam.mx/images/cursos/notas-exportacion.pdf>
- Free Software Foundation Inc. (s.f.). *GNU Public License*. Recuperado el 2 de Mayo de 2013, de <http://www.gnu.org/licenses/gpl.html>
- GIMP. (s.f.). *About Gimp: Introduction*. Recuperado el 3 de Febrero de 2014, de <http://www.gimp.org/about/introduction.html>
- Haines, E. (2016). *Object2Terrain - Inify Community Wiki*. Obtenido de <http://wiki.unity3d.com/index.php?title=Object2Terrain&oldid=18385>
- Hernández Abab, F., Ochoa Vives, M., Hernández Abab, V., & Farré Desongles, C. (2004). *Producción y Tratamiento de Gráficos por Ordenador II* (Segunda ed.). Eureka Media.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2006). *Metodología de la Investigación*. México: McGraw Hill.

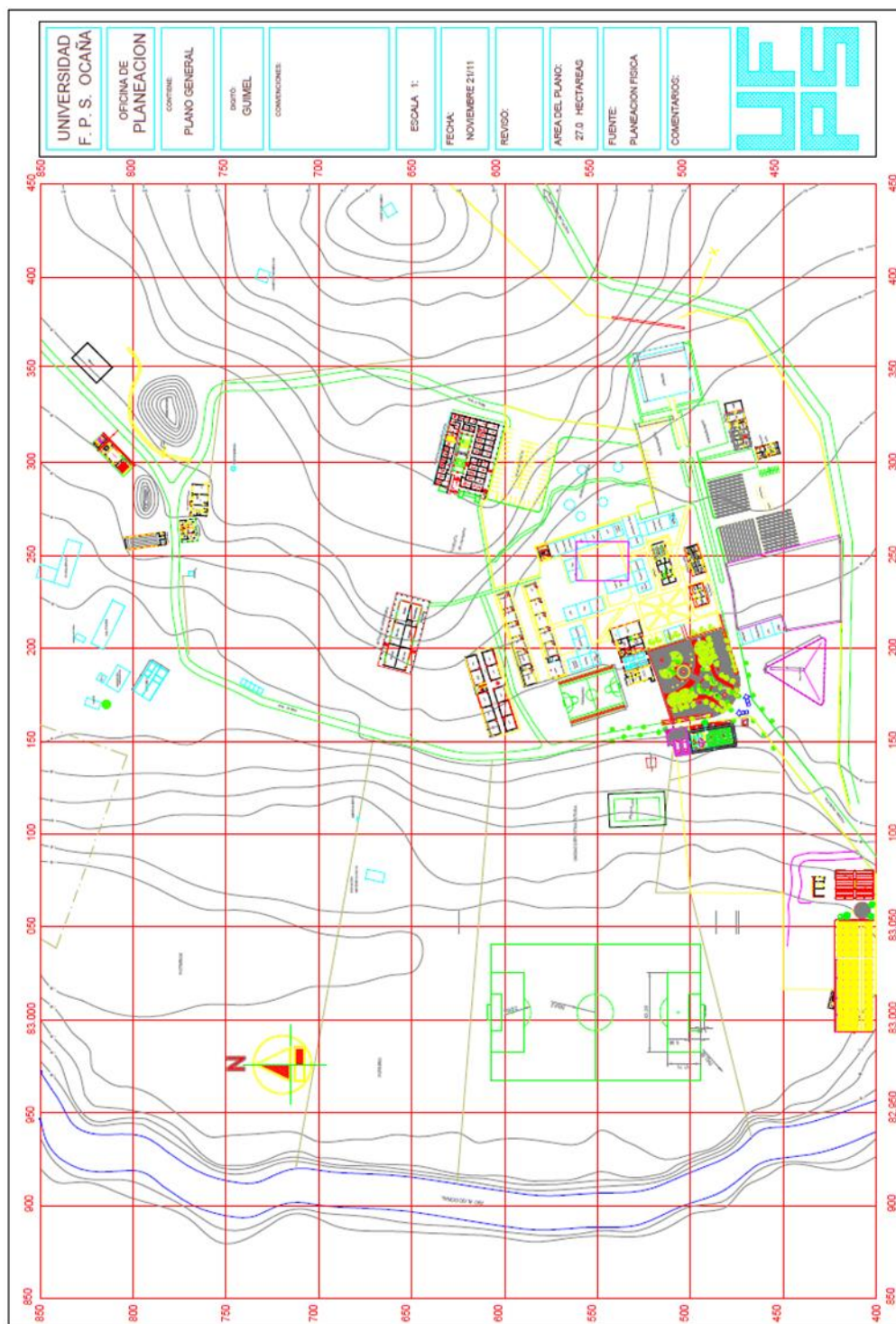
- Leon Guerra, R. (2012). Aplicación de la realidad virtual no inmersiva para Ingenieros Agrícolas. *Rev Cie TécAgr*, 68-72.
- Nadeau, D., Moreland, J., & Heck, Michael M. (s.f.). *Introduction to VRML 97*. Obtenido de <http://www.sdsc.edu/~moreland/courses/Siggraph98/vrml97/slides/mt0006.htm>
- Ros Escribano, V. (2008). Visita virtual, documentación e información de la Escuela Técnica Superior de Ingeniería de Telecomunicación. (*trabajo de grado*). Universidad Politécnica de Cartagena. Escuela Técnica Superior de Ingeniería de Telecomunicación, Cartagena.
- SketchUp. (2015). *SketchUp Make*. Obtenido de <https://www.sketchup.com/es/products/sketchup-make>
- Tamayo, M. (2004). *El Proceso de la Investigación Científica: Incluye evaluación y administración de proyectos de investigación*. México: Limusa.
- U.S. Copyright Office Summary. (1998). *Digital Millenium Copyright Act*. Recuperado el 2 de Mayo de 2013, de <http://www.copyright.gov/legislation/dmca.pdf>
- Ulldemolins, A. (s.f.). Recorridos Virtuales. Fundación para la Universitat Oberta de Catalunya.
- Unity. (s.f.). *Cree Los Juegos Que Le Gustan Con Unity*. Recuperado el 3 de Febrero de 2014, de <http://unity3d.com/es/unity/>
- Unity3D. (2015). *360 Panorama Capture*. Obtenido de <https://www.assetstore.unity3d.com/en/#!/content/38755>
- Unity3D. (2015). *Flujo de trabajo de los Assets (Asset Workflow)*. Obtenido de <http://docs.unity3d.com/es/current/Manual/AssetWorkflow.html>

Unity3D. (s.f.). *Occlusion Culling*. Recuperado el 3 de febrero de 2014, de
<http://docs.unity3d.com/Documentation/Manual/OcclusionCulling.html>

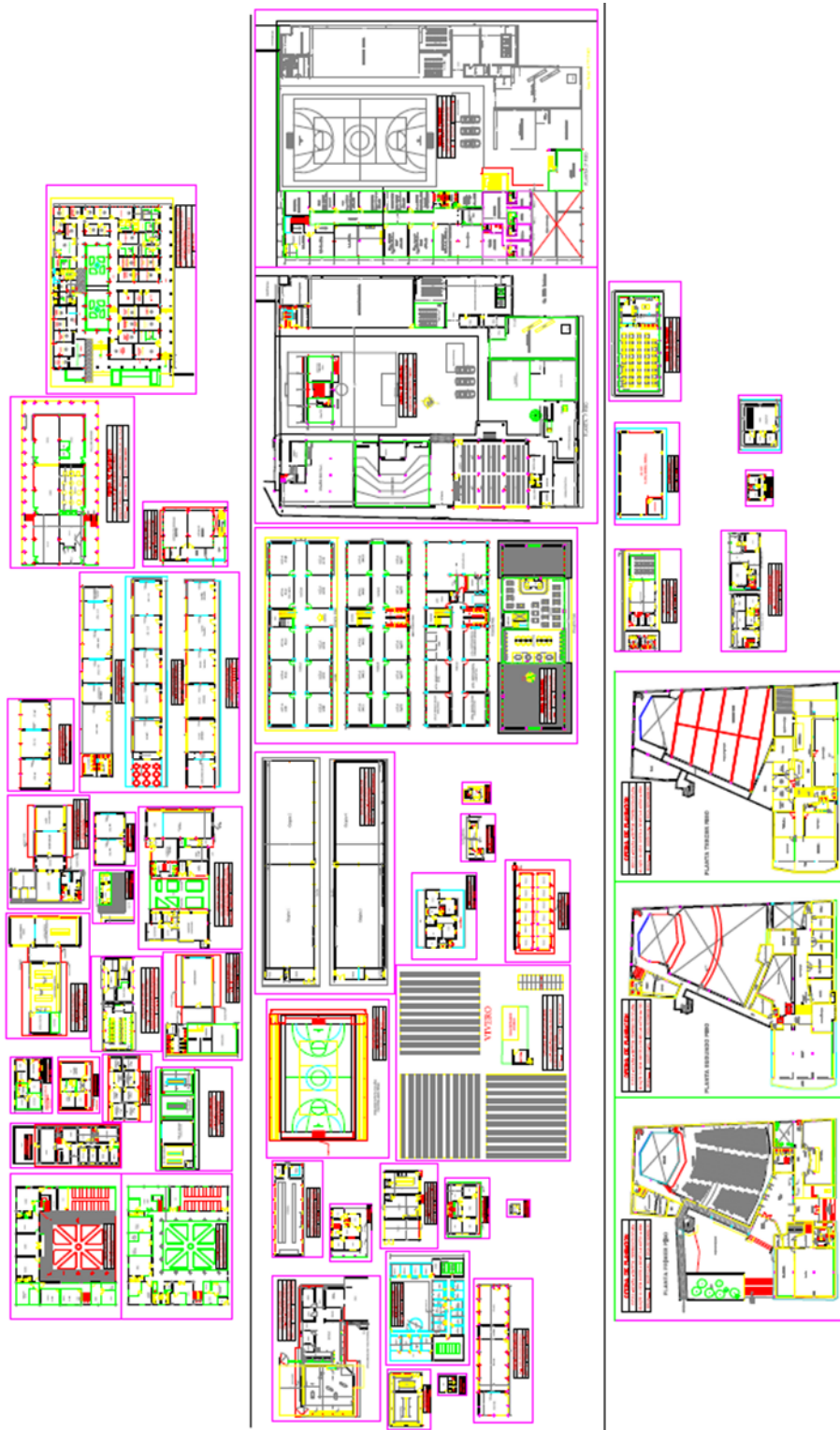
Universidad Francisco de Paula Santander Ocaña. (s.f.). *General*. Obtenido de
<http://www.ufpso.edu.co/ufpso/general.html>

Valencia Martínez, M. A., & Vargas Valencia, D. (2007). Análisis, Diseño e Implementación de un Software para Visitas Virtuales 3D Interactivas por las Instalaciones de la Universidad Tecnológica de Pereira. (*trabajo de grado*). Universidad Tecnológica de Pereira, Pereira.

Apéndices

Apéndice A. Plano General Universidad Francisco de Paula Santander Ocaña.

Apéndice B. Plano por bloques de la Universidad Francisco de Paula Santander Ocaña



Apéndice C. Script Object2Terrain

```

using UnityEngine;
using UnityEditor;

public class Object2Terrain : EditorWindow
{
    [MenuItem("Terrain/Object to Terrain", false, 2000)]
    static void OpenWindow()
    {
        EditorWindow.GetWindow<Object2Terrain>(true);
    }

    private int resolution = 512;
    private Vector3 addTerrain;
    int bottomTopRadioSelected = 0;
    static string[] bottomTopRadio = new string[] { "Bottom Up", "Top Down" };
    private float shiftHeight = 0f;

    void OnGUI()
    {
        resolution = EditorGUILayout.IntField("Resolution", resolution);
        addTerrain = EditorGUILayout.Vector3Field("Add terrain", addTerrain);
        shiftHeight = EditorGUILayout.Slider("Shift height", shiftHeight, -1f, 1f);
        bottomTopRadioSelected = GUILayout.SelectionGrid(bottomTopRadioSelected,
        bottomTopRadio, bottomTopRadio.Length, EditorStyles.radioButton);

        if (GUILayout.Button("Create Terrain"))
        {
            if (Selection.activeGameObject == null)
            {
                EditorUtility.DisplayDialog("No object selected", "Please select an object.", "Ok");
                return;
            }

            else
            {
                CreateTerrain();
            }
        }
    }
}

```

```

    }
}

delegate void CleanUp();

void CreateTerrain()
{
    //fire up the progress bar
    ShowProgressBar(1, 100);

    TerrainData terrain = new TerrainData();
    terrain.heightmapResolution = resolution;
    GameObject terrainObject = Terrain.CreateTerrainGameObject(terrain);

    Undo.RegisterCreatedObjectUndo(terrainObject, "Object to Terrain");

    MeshCollider collider = Selection.activeGameObject.GetComponent<MeshCollider>();
    CleanUp cleanUp = null;

    //Add a collider to our source object if it does not exist.
    //Otherwise raycasting doesn't work.
    if (!collider)
    {
        collider = Selection.activeGameObject.AddComponent<MeshCollider>();
        cleanUp = () => DestroyImmediate(collider);
    }

    Bounds bounds = collider.bounds;
    float sizeFactor = collider.bounds.size.y / (collider.bounds.size.y + addTerrain.y);
    terrain.size = collider.bounds.size + addTerrain;
    bounds.size = new Vector3(terrain.size.x, collider.bounds.size.y, terrain.size.z);

    // Do raycasting samples over the object to see what terrain heights should be
    float[,] heights = new float[terrain.heightmapWidth, terrain.heightmapHeight];
    Ray ray = new Ray(new Vector3(bounds.min.x, bounds.max.y + bounds.size.y,
bounds.min.z), -Vector3.up);
    RaycastHit hit = new RaycastHit();
    float meshHeightInverse = 1 / bounds.size.y;
    Vector3 rayOrigin = ray.origin;

    int maxHeight = heights.GetLength(0);
    int maxLength = heights.GetLength(1);

    Vector2 stepXZ = new Vector2(bounds.size.x / maxLength, bounds.size.z / maxHeight);

```

```

for (int zCount = 0; zCount < maxHeight; zCount++)
{
    ShowProgressBar(zCount, maxHeight);

    for (int xCount = 0; xCount < maxLength; xCount++)
    {
        float height = 0.0f;

        if (collider.Raycast(ray, out hit, bounds.size.y * 3))
        {
            height = (hit.point.y - bounds.min.y) * meshHeightInverse;
            height += shiftHeight;

            //bottom up
            if (bottomTopRadioSelected == 0)
            {
                height *= sizeFactor;
            }

            //clamp
            if (height < 0)
            {
                height = 0;
            }
        }

        heights[zCount, xCount] = height;
        rayOrigin.x += stepXZ[0];
        ray.origin = rayOrigin;
    }

    rayOrigin.z += stepXZ[1];
    rayOrigin.x = bounds.min.x;
    ray.origin = rayOrigin;
}

terrain.SetHeights(0, 0, heights);

EditorUtility.ClearProgressBar();

```

```
    if (cleanUp != null)
    {
        cleanUp();
    }
}

void ShowProgressBar(float progress, float maxProgress)
{
    float p = progress / maxProgress;
    EditorUtility.DisplayProgressBar("Creating Terrain...", Mathf.RoundToInt(p * 100f) + "
%", p);
}
}
```

Apéndice D. Script UnirTerreno

```
using UnityEngine;
using System.Collections;

public class UnirTerreno : MonoBehaviour {

    //Variable publicas que pueden asignarse desde el editor de Unity
    //representan los diferentes terrenos que pueden estar alrededor de
    //un terreno

    public Terrain izq;
    public Terrain arriba;
    public Terrain der;
    public Terrain abajo;

    void Start () {

        //Obtiene la referencia al componente terreno del objeto actual.
        Terrain terrenoAct = gameObject.GetComponent<Terrain>();

        //Asigna los terrenos vecinos al terreno actual.
        terrenoAct.SetNeighbors( izq, arriba, der, abajo);

    }

}
```


Apéndice E. Script AdminAssetBundle

```

using UnityEngine;
using System.Collections;

public class AdminAssetBundle : MonoBehaviour {

    public string dirServAssetBundle;
    public string[] nombresAssetBundles;

    void Start ()
    {
        int numAssetBundles = nombresAssetBundles.Length;
        if (numAssetBundles > 0) {

            string[] urlAssetBundles = new string[numAssetBundles];

            for (int i = 0; i < numAssetBundles; i++) {

                if (dirServAssetBundle.EndsWith ("/")) {
                    urlAssetBundles [i] = dirServAssetBundle +
nombresAssetBundles [i];
                } else {
                    urlAssetBundles [i] = dirServAssetBundle + "/" +
nombresAssetBundles [i];
                }

            }

            StartCoroutine(descargarBundles(urlAssetBundles) );
        }
    }

    IEnumerator descargarBundles(string[] urlBundle){

        for (int i = 0; i < urlBundle.Length; i++) {
            while (!Caching.ready)
                yield return null;
            WWW descarga;
            try
            {
                descarga = WWW.LoadFromCacheOrDownload(urlBundle[i], 0);
            }
            catch (System.Exception)
            {

```

```
        throw;
    }

    yield return descarga;

    if (descarga.error != null) {
        Debug.LogError (descarga.error);
    }

    var objetos = descarga.assetBundle;
    objetos.LoadAllAssets();
    descarga.Dispose();
}
}
```

Apéndice F. Scripts para controlar la visibilidad de los objetos dentro del recorrido

Script TriggerPadre.cs

```
using UnityEngine;
using System.Collections;

public class TriggerPadre : MonoBehaviour {

    public int activaciones;
    void Start(){
        activaciones = 0;
        Debug.Log("Inicializado padre " + gameObject.name);
    }
}
```

Script TriggerRenderAne.cs

```
using UnityEngine;
using System.Collections;

public class TriggerRenderAne : MonoBehaviour {

    TriggerPadre padre;
    GameObject hijo;

    void Start() {
        padre = GetComponentInParent<TriggerPadre>() as TriggerPadre;
        string pad = transform.parent.name;
        hijo = GameObject.Find("Anexos").transform.Find(pad).gameObject;
        hijo.SetActive (false);
        Debug.Log("Listo trigger: " + gameObject.name);
    }

    void OnTriggerEnter(Collider other) {
        padre.activaciones += 1;
        hijo.SetActive (true);
        Debug.Log (gameObject.name + "Activando " + hijo.name);
    }
}
```

```
void OnTriggerExit(Collider other) {
    padre.activaciones -= 1;
    if(padre.activaciones == 0){
        hijo.SetActive(false);
        Debug.Log(gameObject.name + "Desactivando " + hijo.name);
    }
}
}
```

Apéndice G. Libreto del dialogo de Santanderito

Bienvenida

¡Hola!

Soy Santanderito.

Te doy la bienvenida a la Universidad Francisco de Paula Santander seccional Ocaña.

Controles

Puedes saltar una parte de la introducción haciendo clic en el botón “Omitir” en la esquina inferior izquierda, o puedes saltar todo el diálogo haciendo clic en el botón “salir”.

Historia

Ahora te haré una breve reseña para que conozcas un poco sobre nuestra Universidad:

La Universidad Francisco de Paula Santander Seccional Ocaña se crea según Acuerdo número 003 del 18 de Julio de 1974, por parte del Consejo Superior de la Universidad Francisco de Paula Santander Cúcuta, como máxima expresión cultural y patrimonio de la región.

Inicialmente tuvo como sede los claustros Franciscanos al costado del templo de la Gran Convención. En 1975 comenzó la actividad académica en la entonces seccional de la Universidad Francisco de Paula Santander con un total de 105 estudiantes de Tecnología en Matemáticas y Física.

La consecución de la Hacienda El Rhin, en las riveras del Río Algodonal, en comodato, que la antigua Escuela de Agricultura de Ocaña cedió a la Universidad, permitió la creación del programa de Tecnología en Producción Agropecuaria, aprobado por el Consejo Superior el 21 de agosto de 1980, y luego el ICFES otorgó la licencia de funcionamiento el 17 de febrero del año siguiente.

Nuestra Universidad ha ido creciendo tras la creación de nuevas facultades, contando en este momento con la Facultad de Ciencias Agrarias y del Ambiente; Facultad de Ciencias Administrativas y Económicas; Facultad de Ingenierías y la Facultad de Educación, Artes y Humanidades.

Contenido

Ahora que conoces un poco más de nuestra universidad, te invito a visitar los edificios Casona y Anexos; pero antes que todo te daré una breve introducción de lo que encontraras.

En el edificio Casona podrás encontrar toda la parte administrativa de nuestra Universidad, además del auditorio Fabio Amaya; en esta sección del campus se encuentran las oficina de: dirección, secretaría, control interno, planeación, subdirección administrativa, subdirección académica, división de personal, unidad de presupuesto, contabilidad, tesorería, la oficina de admisiones, registro y control y la división de investigación y extensión.

A diferencia de la casona, en Anexos, tenemos la parte académica de pregrados, contando con las facultades de: Ciencias Agrarias y del Ambiente, Ciencias Administrativas y Económicas, Ingenierías y la Facultad de Educación, Artes y Humanidades; además de los planes de estudio de: Ingeniería Ambiental, Zootecnia, Administración de Empresas, Contaduría

Pública, Ingeniería Civil, Ingeniería de Sistemas, Ingeniería Mecánica, Comunicación Social y las Tecnologías y Técnicos en: Tecnología Agropecuaria, Tecnología en Gestión Comercial y Financiera, Tecnología en Gestión de Mercado, Tecnología en Gestión Empresarial Agropecuaria, Tecnología en Gestión Financiera, Tecnología en Gestión Empresarial de Economía Solidaria, Técnico Profesional en Gestión Comercial y Financiera, Técnico Profesional en Gestión de Mercadeo, Técnico Profesional en Empresas Agropecuarias, Técnico Profesional en Gestión Empresarial en Economía Solidaria, Técnico Profesional en Telecomunicaciones, Técnico Profesional en Informática.

Explicación

Ahora sí, el momento ha llegado, recuerda que para navegar en el entorno deberás hacer uso de tu mouse y teclado, con las flechas de dirección podrás avanzar por la escena y con el mouse podrás controlar la cámara. Esperamos que disfrutes tu recorrido.

Apéndice H. Scripts para generar animaciones de la sincronización labial de Santanderito.

Script ModEditorAni.cs

```
using UnityEngine;
using System.Collections;
using UnityEditor;

[CustomEditor(typeof(Animacion))]
public class ModEditorAni : Editor
{
    public override void OnInspectorGUI()
    {
        DrawDefaultInspector();

        Animacion myScript = (Animacion)target;

        if (GUILayout.Button("Crear Animacion"))
        {
            myScript.iniciar();
        }
    }
}
```

Script Animacion.cs

```
using UnityEngine;
using System.Collections;
using UnityEditor;

public class Animacion : MonoBehaviour
{
    SkinnedMeshRenderer skinnedMeshRenderer;
    Mesh skinnedMesh;
    string nombreAnimacion;
    public float FramesXSeg = 24f;
    public float frameRate= 24f;
    public TextAsset textAsset;
    public float fuerzaBlendShape = 50.0f;
    public float porcentajeFuerza = 50f;
    public float porcentajeTiempo = 50f;
```



```

int numBlenshape;
Transform hijo;
string ruta;
public int frameAnt = 3;    //frames de apertura
public int framePos = 3;    //frames de cierre
public int tangentMode= 0; //modo de la tangente
string[] visemas;
float[] tiempo;
public int indiceAudio = 0;
float tiempoKeyFrame;
bool[] curvasUsadas;
Keyframe keyframe;
AnimationCurve[] curvas;

public void iniciar()
{

    nombreAnimacion = textAsset.name;
    hijo = transform.FindChild("Santanderito");
    ruta = hijo.name;
        skinnedMeshRenderer = hijo.GetComponent<SkinnedMeshRenderer>();
    skinnedMesh = hijo.GetComponent<SkinnedMeshRenderer>().sharedMesh;
    numBlenshape = skinnedMesh.blendShapeCount;
    animar();
}

void prepararVisemas()
{
    string[] separador = { "\n" };
        string[] lista = textAsset.text.Split(separador, System.StringSplitOptions.None);
        visemas = new string[lista.Length];
        tiempo = new float[lista.Length];
        separador[0] = " ";
        for (int i = 0; i < tiempo.Length; i++) {
            visemas[i] = lista[i].Split(separador,
System.StringSplitOptions.None)[1].Trim();
            tiempo[i]= float.Parse(lista [i].Split (separador,
System.StringSplitOptions.None)[0].Trim ()) / FramesXSeg;
        }
}

void animar()
{
    AnimationClip clip = new AnimationClip();
        clip.frameRate = frameRate;
}

```

```

curvas = new AnimationCurve[numBlenshape];

tiempoKeyFrame = 1.0f / FramesXSeg;
curvasUsadas = new bool[numBlenshape];
    prepararVisemas ();

for (int i = 0; i < numBlenshape; i++)
{
    curvas[i] = new AnimationCurve();
    curvasUsadas[i] = false;
        keyframe = new Keyframe(0f, 0f, 0.0f, 0.0f);
        curvas[i].AddKey(keyframe);
}

    for (int i = 0; i < visemas.Length; i++)
    { //si hay dos seguidas, rest
        if (visemas [i] == "rest") {
            for (int j = 0; j < numBlenshape; j++)
            {
                keyframe = new Keyframe(0f, 0f, 0.0f, 0.0f);
                curvas[j].AddKey(keyframe);
            }
        }
        else if (((i+1) < visemas.Length) && (visemas[i] == visemas[i+1])){
            //asignamos la entrada al visema actual
            keyframe = new Keyframe(tiempo[i]-
(tiempoKeyFrame*frameAnt), 0f, 0.0f, 0.0f);
            keyframe.tangentMode = tangentMode;

            curvas[skinnedMesh.GetBlendShapeIndex(visemas[i])].AddKey(keyframe);
            //asignamos el visema actual
            keyframe = new Keyframe(tiempo[i], fuerzaBlendShape, 0.0f,
0.0f);

            keyframe.tangentMode = tangentMode;

            curvas[skinnedMesh.GetBlendShapeIndex(visemas[i])].AddKey(keyframe);
            //se ajusta la transicion entre los visemas
            keyframe = new Keyframe(((tiempo[i]-
tiempo[i+1])*(porcentajeTiempo/100))+tiempo[i], fuerzaBlendShape*(porcentajeFuerza/100),
0.0f, 0.0f); //porcentaje de tiempo?
            keyframe.tangentMode = tangentMode;

            curvas[skinnedMesh.GetBlendShapeIndex(visemas[i])].AddKey(keyframe);
            //asignamos el siguiente visema
            keyframe = new Keyframe(tiempo[i+1], fuerzaBlendShape, 0.0f,
0.0f);

```

```

        keyframe.tangentMode = tangentMode;

        curvas[skinnedMesh.GetBlendShapeIndex(visemas[i+1])].AddKey(keyframe);
        //asignamos la salida del siguiente visema
        keyframe = new
Keyframe(tiempo[i+1]+(tiempoKeyFrame*framePos), 0f, 0.0f, 0.0f);
        keyframe.tangentMode = tangentMode;

        curvas[skinnedMesh.GetBlendShapeIndex(visemas[i+1])].AddKey(keyframe);
        i++;
    }
    else {
        //asignamos la entrada al visema actual
        keyframe = new Keyframe(tiempo[i]-
(tiempoKeyFrame*frameAnt), 0f, 0.0f, 0.0f);
        keyframe.tangentMode = tangentMode;

        curvas[skinnedMesh.GetBlendShapeIndex(visemas[i])].AddKey(keyframe);
        //asignamos el visema actual
        keyframe = new Keyframe(tiempo[i], fuerzaBlendShape, 0.0f,
0.0f);
        keyframe.tangentMode = tangentMode;

        curvas[skinnedMesh.GetBlendShapeIndex(visemas[i])].AddKey(keyframe);
        //asignamos la salida del visema actual
        keyframe = new
Keyframe(tiempo[i]+(tiempoKeyFrame*framePos), 0f, 0.0f, 0.0f);
        keyframe.tangentMode = tangentMode;

        curvas[skinnedMesh.GetBlendShapeIndex(visemas[i])].AddKey(keyframe);
        //se marca las curvas utilizadas en la animacion
        curvasUsadas [skinnedMesh.GetBlendShapeIndex (visemas [i])] =
true;
    }
}

    blink ();
    AnimationEvent[] aniEvento = new AnimationEvent[1];
    aniEvento[0] = new AnimationEvent ();
    aniEvento[0].functionName = "repAudio1";
    aniEvento[0].intParameter = indiceAudio;
    aniEvento[0].time = 0f;
    AnimationUtility.SetAnimationEvents (clip, aniEvento);
for (int i = 0; i < numBlenshape; i++)
{
    if (curvasUsadas[i])
    {

```

```

        AnimationUtility.SetEditorCurve(clip, EditorCurveBinding.FloatCurve(ruta,
typeof(SkinnedMeshRenderer), "blendShape." + skinnedMesh.GetBlendShapeName(i)),
curvas[i]);
    }

}

guardarAnimacion(clip, nombreAnimacion);

}

void guardarAnimacion(AnimationClip clipAnimacion, string nomAnimacion)
{
    string ANIM_CLIP_PATH = "Assets/Animaciones/";
    AssetDatabase.CreateAsset(clipAnimacion, ANIM_CLIP_PATH + nomAnimacion +
".anim");
    AssetDatabase.SaveAssets();
}

void blink()
{
    int veces = (int)Random.Range (1f, tiempo [tiempo.Length - 1] * 0.2f);
    float tiempoIni = 1.5f;
    float tiempoBlink;
    for (int i = 0; i < veces; i++) {

        tiempoBlink = Random.Range(tiempoIni, (tiempo[tiempo.Length-1]));

        keyframe = new Keyframe(tiempoBlink-0.15f, 0f, 0.0f, 0.0f);
        keyframe.tangentMode = 0;
        curvas[skinnedMesh.GetBlendShapeIndex("blink")].AddKey(keyframe);

        keyframe = new Keyframe(tiempoBlink, 100f, 0.0f, 0.0f);
        keyframe.tangentMode = 0;
        curvas[skinnedMesh.GetBlendShapeIndex("blink")].AddKey(keyframe);

        keyframe = new Keyframe(tiempoBlink+0.15f, 0f, 0.0f, 0.0f);
        keyframe.tangentMode = 0;
        curvas[skinnedMesh.GetBlendShapeIndex("blink")].AddKey(keyframe);

        tiempoIni = tiempoBlink;
    }
    curvasUsadas [skinnedMesh.GetBlendShapeIndex ("blink")] = true;
}
}

```



```

private var anguloRotacion : float = 0;
private var mitadAnchoVentanaMiniMapa = anchoVentanaMiniMapa * 0.5;
private var mitadAltoVentanaMiniMapa = altoVentanaMiniMapa * 0.5;
private var pivotPoint : Vector2 = Vector2(mitadAnchoVentanaMiniMapa, mitadAltoVentanaMiniMapa);
private var posXPunto:float;
private var posYPunto:float;
private var mapScaleX:float;
private var mapScaleY:float;
private var mostrarMiniMapa = false;
private var tiempoInicio : double;
private var matrixBackup : Matrix4x4;
private var puntoYBotones = altoVentanaMiniMapa - 20;
private var puntoXBotonAcercar = mitadAnchoVentanaMiniMapa - 20;
private var puntoXBotonAlejar = mitadAnchoVentanaMiniMapa;
private var minZoom = 256; //pixeles
private var maxZoom = 1024; //pixeles
private var maximizar = false;

function mostrarMMapa(){
    mostrarMiniMapa = !mostrarMiniMapa;
}

function escalaMiniMapa()
{

    mapScaleX = parseFloat(anchoMiniMap) / anchoTerreno;
    mapScaleY = parseFloat(altoMiniMap) / altoTerreno;

}

function Start()
{
    tiempoInicio = Time.realtimeSinceStartup;
    posicionPunto();
    escalaMiniMapa();
}

function posicionPunto()
{
    posXPunto = mitadAnchoVentanaMiniMapa - (anchoPunto * 0.5);
    posYPunto = mitadAltoVentanaMiniMapa - (altoPunto * 0.5);
}

function OnGUI()
{

```

```

if (Input.GetKeyUp(KeyCode.M))
{
    var tiempoActual:double = Time.realtimeSinceStartup;
    if((tiempoActual-tiempoInicio)>0.1)
    {
        if(!maximizar )
            maximizar = true;
        else
            maximizar = false;
        tiempoInicio = tiempoActual;
    }
}
if(mostrarMiniMapa)
    ventanaMiniMapa = GUI.Window (0, ventanaMiniMapa, DoMyWindow, "Recorrido Ufp
so");
}

function miniMapa()
{
    var pX = (objetoCentro.position.x - offsetX) * mapScaleX;
    var pY = (objetoCentro.position.z - offsetY) * mapScaleY;
    GUI.DrawTexture(Rect(0, 0, ventanaMiniMapa.width, ventanaMiniMapa.height), fondoMini
Mapa);
    GUI.DrawTexture(Rect(mitadAnchoVentanaMiniMapa - pX, mitadAltoVentanaMiniMapa -
(altoMiniMap-pY), anchoMiniMap, altoMiniMap), imagenMinMapa);
    GUI.DrawTexture(Rect(0, 0, ventanaMiniMapa.width, ventanaMiniMapa.height), mascaraMi
niMapa);
    anguloRotacion = getAnguloObjetoCentral();
    matrixBackup = GUI.matrix;
    GUIUtility.RotateAroundPivot (anguloRotacion, pivotPoint);
    GUI.DrawTexture(Rect(posXPunto, posYPunto, anchoPunto, altoPunto), puntoMapa);
    GUI.matrix = matrixBackup;
}

function zoomMiniMap()
{
    if (GUI.RepeatButton(Rect(puntoXBotonAcercar,puntoYBotones,20,20),"+"))
    {
        if(anchoMiniMap >= maxZoom || altoMiniMap >= maxZoom )
            return;
        anchoMiniMap+=5;
        altoMiniMap+=5;
        escalaMiniMapa();
    }
    if (GUI.RepeatButton(Rect(puntoXBotonAlejar,puntoYBotones,20,20),"-"))
    {

```

```

    if(anchoMiniMap < minZoom || altoMiniMap < minZoom )
        return;
    anchoMiniMap-=5;
    altoMiniMap-=5;
    escalaMiniMapa();
}
}

function DoMyWindow (windowID : int) {
    ajustarMiniMapa();
    escalaMiniMapa();
    miniMapa();
    zoomMiniMap();
    GUI.DragWindow();
}

function getAnguloObjetoCentral()
{
    return objetoCentro.eulerAngles.y;
}

function ajustarMiniMapa()
{
    if(tamannoVent!=Screen.width * 0.25 || maximizar){
        if(!maximizar)
            tamannoVent = Screen.width * 0.25;
        else
            tamannoVent = Screen.height * 0.9;
        anchoVentanaMiniMapa = tamannoVent;
        altoVentanaMiniMapa = tamannoVent ;
        mitadAnchoVentanaMiniMapa = tamannoVent/2;
        mitadAltoVentanaMiniMapa = mitadAnchoVentanaMiniMapa;
        posicionPunto();
        pivotPoint = Vector2(mitadAnchoVentanaMiniMapa, mitadAltoVentanaMiniMapa);
        if(!maximizar){
            ventanaMiniMapa = Rect ((Screen.width-tamannoVent),
                (Screen.height-tamannoVent),
                tamannoVent,
                tamannoVent);
            puntoYBotones = tamannoVent - 20;

            puntoXBotonAcercar = mitadAnchoVentanaMiniMapa - 20;
            puntoXBotonAlejar = mitadAnchoVentanaMiniMapa;
        }else{
            var offsetmax = Screen.height * 0.9;
            ventanaMiniMapa = Rect (((Screen.width-offsetmax)/2),

```



```
                ((Screen.height-offsetmax)/2),
                tamannoVent,
                tamannoVent);
puntoYBotones = tamannoVent - 20;

puntoXBotonAcercar = mitadAnchoVentanaMiniMapa - 20;
puntoXBotonAlejar = mitadAnchoVentanaMiniMapa;
    }
    yield;
}
}
```