

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE SOFTWARE

TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN
DEL TÍTULO DE INGENIERO EN SOFTWARE

TEMA:

**“Desarrollo de un tour virtual “Paseo UTN” con Realidad
Aumentada para la Universidad Técnica del Norte usando SCRUM y
la ISO/IEC 25022”**



AUTOR: Victor Eduardo Echeverría Soto

DIRECTOR: Cathy Pamela Guevara Vega

Ibarra-Ecuador

2026

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional para lo cual pongo a disposición la siguiente información

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004096192		
APELLIDOS Y NOMBRES:	Echeverría Soto Victor Eduardo		
DIRECCIÓN:	Imbabura – Ibarra, Quilago 5-16 y Av. El Retorno		
EMAIL:	veecheverrias@utn.edu.ec		
TELÉFONO FIJO:	-----	TELF. MOVIL	0993376288

DATOS DE LA OBRA	
TÍTULO:	Desarrollo de un tour virtual “Paseo UTN” con Realidad Aumentada para la Universidad Técnica del Norte usando SCRUM y la ISO/IEC 25022
AUTOR (ES):	Victor Eduardo Echeverría Soto
FECHA: DD/MM/AA	18/02/2026
CARRERA/PROGRAMA:	☒ PREGRADO ☐ POSGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SOFTAWRE
DIRECTOR:	Ing. Cathy Pamela Guevara Vega, PhD.
ASESOR	Ing. José Antonio Quiña Mera, PhD.

CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 18 días, del mes de febrero de 2026

EL AUTOR:

Firma.....

Nombre: Victor Eduardo Echeverría Soto

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 18 de febrero de 2026

Ing. Cathy Pamela Guevara Vega, PhD.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

(f)
Cathy Pamela Guevara Vega
C.C.: 1002334835

DEDICATORIA

Quiero dedicar este trabajo a mi padre y madre. Ellos fueron quienes me han dado todo en mi vida, todos mis logros se los debo a ellos porque nunca me faltó la comida, el transporte, la ropa, un techo, los medios para estudiar y trabajar. Nunca me faltó nada gracias a ellos. Soy quien soy ahora por ellos.

A mi padre, Jorge Echeverría. Has sido el pilar de mi vida, mi definición de fuerza, mi apoyo incondicional y quien siempre ha estado de mi lado. Sin ti no habría llegado tan lejos. Gracias por todo.

A mi madre, Gladys Soto, mi querida madre quien le da color al hogar, gracias por ayudarme, escucharme y cuidarme. De ti aprendí lo que significa el amor incondicional, que la familia es importante y que sin importar qué, estarás para mí.

AGRADECIMIENTO

Quiero agradecer principalmente a mi querida Universidad Técnica del Norte, donde aprendí y reforcé mis principios, me ayudó a crecer y me permitió conocer personas increíbles además de brindarme grandes oportunidades que no se repiten en la vida.

Agradezco al Ing. Antonio Quiña y a la Ing. Cathy Guevara por haberme brindado la oportunidad de ser parte de un proyecto tan grande y apoyarme guiándome y aprovechando mi potencial.

Ing. Carlos Vásquez muchas gracias por haber sido mi mentor y guía. Usted me demostró la forma correcta de hacer las cosas y gracias a usted puedo ser una mejor persona y profesional.

Eduardo Díaz y Esmeralda Pérez, les agradezco mucho por haber aceptado formar parte del equipo y haberse esforzado tanto por dar un resultado al nivel.

Mis compañeros y amigos, de no ser por ustedes la universidad se habría vuelto monótona, gracias por haber hecho de mis días en la universidad un lugar lleno de color y de recuerdos.

Agradezco de manera especial a:

Alan Sandoval
Bryan de la Cruz
Juliana Abarca

Gracias por el apoyo brindado durante el desarrollo de este trabajo.

RESUMEN EJECUTIVO

El presente trabajo se desarrolla debido a la necesidad de facilitar la orientación y el reconocimiento de espacios físicos dentro del campus universitario, ya que muchos estudiantes y visitantes tienen dificultades para ubicar edificios, dependencias y puntos importantes cuando visitan la universidad. El objetivo general fue desarrollar e implementar la aplicación móvil “Paseo UTN”, la cual permite explorar un modelo 3D del campus, visualizar información de edificios y alternar entre modos de visualización, integrando además una solución complementaria para la administración de contenidos. La metodología SCRUM permitió un desarrollo iterativo de 4 sprints en donde se utilizó Unity como marco de trabajo, Blender para el modelo 3D y la incorporación de funciones de interacción, navegación e interfaz de usuario. La evaluación se desarrolló mediante un taller práctico con ocho tareas, registrando finalización, errores y tiempos de ejecución, junto con un cuestionario tipo Likert de percepción del usuario, ajustando las preguntas negativas para mantener coherencia en la interpretación de resultados. Las métricas usadas en la evaluación fueron eficacia, eficiencia y satisfacción basadas en la norma de calidad en uso ISO/IEC 25022. Los resultados evidenciaron un alto nivel de eficacia, con una finalización promedio aproximada del 96.6%, una baja tasa de errores concentrada principalmente en el cambio de modo a realidad aumentada, un tiempo total promedio de ejecución de 234.66 segundos, y una satisfacción global corregida de 4.012/5. Finalmente, la integración ponderada de eficacia, eficiencia y satisfacción permitió obtener un valor global de calidad en uso cercano al 86.2%, lo que demuestra que “Paseo UTN” constituye una alternativa funcional y útil para mejorar la exploración del campus y apoyar la ubicación de espacios universitarios.

Palabras clave: Realidad Aumentada, Modelo 3D, Orientación Universitaria, Unity, ISO/IEC 25022, AR Foundation.

ABSTRACT

This work was developed due to the need to facilitate orientation and space recognition within the university campus, since many students and visitors have difficulties locating buildings, offices, and key points without in-person assistance. The main objective was to design and implement the mobile application “Paseo UTN”, which allows users to explore a 3D campus model, view information about buildings, and switch between visualization modes, while also including a complementary solution for content management. The methodology followed an iterative development approach using Unity, a modular 3D campus model, and the implementation of interaction, navigation, and user interface features, followed by an evaluation focused on quality in use based on ISO/IEC 25022. The assessment included a practical task-based workshop with eight tasks, recording task completion, errors, and execution times, together with a Likert-scale user perception questionnaire, adjusting negatively worded items to ensure consistent interpretation of results. The findings showed a high level of effectiveness, with an average task completion rate of approximately 96.6%, a low error rate mainly associated with switching to AR mode, an average total execution time of 234.66 seconds, and a corrected overall satisfaction score of 4.012/5. Finally, the weighted integration of effectiveness, efficiency, and satisfaction produced an overall quality in use value close to 86.2%, showing that “Paseo UTN” is a functional and useful alternative to improve campus exploration and support the location of university spaces.

Keywords: Augmented Reality, 3D Model, University Orientation, Unity, ISO/IEC 25022, AR Foundation.

ÍNDICE DE CONTENIDOS

CAPÍTULO I INTRODUCCIÓN.....	1
1.1 Tema	1
1.2 Problema de investigación	1
1.3 Objetivos	2
1.3.1 Objetivo General	2
1.3.2 Objetivos Específicos.....	2
1.4 Alcance	2
1.5 Metodología	3
1.6 Justificación	4
1.7 Antecedentes	5
CAPÍTULO II MARCO TEÓRICO	7
2.1 Conceptos Fundamentales	7
2.1.1 Realidad Aumentada (RA).....	7
2.1.2 Modelado 3D.....	10
2.1.3 Tours Virtuales.....	12
2.2 Tecnologías y Herramientas.	13
2.2.1 Unity y Realidad aumentada.	13
2.2.2 Blender.	15
2.2.3 Metodología SCRUM.	15
2.2.4 ISO/IEC 25022.....	17
2.3 Trabajos Relacionados.....	18
2.5.1 Virtual Campus Tour Application through Markerless Augmented Reality Approach.....	18
2.5.2 Virtual Campus Tour Berbasis Web Menggunakan Teknologi Augmented Reality.....	18
2.5.3 Aplikasi Virtual Tour Islamic Center Berbasis Android.....	18

2.5.4 Memórias Tridimensionais do Grande ABC: Tour Virtual 3D pela Vila de Paranapiacaba.	19
2.5.5 Virtual Tour With Voice Assistant using Extended Reality.	19
2.5.6 Virtual Reality Based Virtual Tour of College Using Unity 3D.....	19
2.5.7 Implementing Augmented reality in Tourism.	20
2.5.8 Exploración en Arquitectura: Un proceso basado en Blender	20
CAPÍTULO III DESARROLLO	21
3.1 Fase: Pre-juego	21
3.1.1 Definición de roles para el desarrollo de la aplicación:	21
3.1.2 Elicitación de requisitos	22
3.1.3 Product Backlog	31
3.2 Fase: Juego.....	32
3.2.1 Sprint 1 - Visualización base y modos de interacción	32
3.2.2 Sprint 2 – Interfaz de usuario y navegación principal.....	36
3.2.3 Sprint 3 – Implementación del modo Street View	42
3.2.4 Sprint 4 – Optimización, seguridad y preparación para despliegue...	45
3.3 Fase: Post-juego	49
3.3.1 Despliegue del producto.....	50
3.3.2 Verificación de funcionamiento	50
3.3.3 Entrega del producto	51
3.3.4 Cierre de la fase pos-juego	51
CAPÍTULO IV RESULTADOS Y ANÁLISIS	51
4.1 Modelo de evaluación aplicado (ISO/IEC 25022 – Calidad en uso).....	51
4.2 Población y muestra.....	53
4.3 Resultados del Taller práctico.....	53
4.3.1 Eficacia.....	53
4.3.2 Errores durante la ejecución	54

4.3.3 Eficiencia temporal	55
4.4 Resultados del cuestionario.....	55
4.4.1 Resultados por pregunta (Likert 1–5).....	56
4.4.2 Resultado por subcaracterística	57
4.5 Validación del instrumento y confiabilidad.....	57
4.5.1 Prueba de normalidad.....	58
4.5.2 Confiabilidad del cuestionario (Alfa de Cronbach)	58
4.6 Resultado global de Calidad en Uso	58
4.7 Análisis e interpretación general.....	59
Conclusiones y recomendaciones	60
Conclusiones	60
Recomendaciones	60
Referencias Bibliográficas.....	61
Anexos.....	63
Anexo A. Fotografías de la aplicación del taller práctico y encuesta SUS.....	63
Anexo B. Presentación de proyectos y equipo UTN Móvil a la máxima autoridad UTN.	65
Anexo C. Reuniones equipo UTN Móvil	66
Anexo D. Presentación de UTN Móvil a un curso	68
Anexo E. Taller práctico aplicado en la evaluación del aplicativo Paseo UTN	69
Anexo F. Tabulación de los datos obtenidos del taller práctico aplicado en la evaluación del aplicativo Paseo UTN	69
Anexo G. Manual Técnico de la aplicación.....	71

ÍNDICE DE TABLAS

Tabla 1 Antecedentes del Proyecto	5
Tabla 2 Roles de SCRUM	21
Tabla 3 Historia de Usuario 1	22
Tabla 4 Historia de Usuario 2	22
Tabla 5 Historia de Usuario 3	23
Tabla 6 Historia de Usuario 4	23
Tabla 7 Historia de Usuario 5	24
Tabla 8 Historia de Usuario 6	24
Tabla 9 Historia de Usuario 7	25
Tabla 10 Historia de Usuario 8	25
Tabla 11 Historia de Usuario 9	26
Tabla 12 Historia de Usuario 10	26
Tabla 13 Historia de Usuario 11	26
Tabla 14 Historia de Usuario 12	27
Tabla 15 Historia de Usuario 13	27
Tabla 16 Historia de Usuario 14	28
Tabla 17 Historia de Usuario 15	28
Tabla 18 Historia de Usuario 16	28
Tabla 19 Historia de Usuario 17	29
Tabla 20 Historia de Usuario 18	30
Tabla 21 Historia de Usuario 19	30
Tabla 22 Modelo de evaluación basado en ISO/IEC 25022	52

ÍNDICE DE FIGURAS

Figura 1. Árbol de problemas	2
Figura 2 Relación de componentes.....	3
Figura 3 Ejemplo de RA con imágenes [8].	9
Figura 4 Realidad aumentada con geolocalización [10].....	9
Figura 5 Realidad aumentada con Google Glasses [10].....	10
Figura 6 Ejemplo de niveles de detalle (LOD) [13].	11
Figura 7 Comparativa de cantidad de polígonos en videojuegos distintos [11].	12
Figura 8 Escena en el editor de Unity [17].	13
Figura 9 Flujo de trabajo de assets en Unity 3D [18].	14
Figura 10 Escena de Realidad Aumentada en el Editor de Unity [19].	14
Figura 11 Imagen renderizada y procesada con Blender [20].	15
Figura 12 Fases de SCRUM [34].	21
Figura 13 Modelo 3D de la UTN realizado con Fotogrametría (15 millones de polígonos)	33
Figura 14 Modelo 3D de la UTN placeholder	34
Figura 15 Modelo 3D UTN v1 realizado en colaboración tesistas de Diseño Gráfico	34
Figura 16 Modelo 3D v5 realizado en colaboración tesistas de Diseño Gráfico	35
Figura 17 Modelo 3D de la UTN en escenario de modo NoAR	36
Figura 18 Vista de edificios en el modo NoAR.....	36
Figura 19 Interfaz del modo con Realidad Aumentada (AR) Vertical.....	37
Figura 20 Interfaz del modo sin Realidad Aumentada (NoAR) Vertical	38
Figura 21 Interfaz del modo con Realidad Aumentada (AR) Horizontal.....	38
Figura 22 Interfaz del modo sin Realidad Aumentada (NoAR) Horizontal.....	39
Figura 23 Interfaz del modo Street View (SV).....	39
Figura 24 Opciones del menú en interfaz	40
Figura 25 Botones de interacción (Captura, Ingreso modo SV, Información)...	40
Figura 26 Guía de gestos en pantalla de carga	41
Figura 27 Ventana de información	42
Figura 28 Cámara en modo Street View (SV) en editor de Unity.....	43
Figura 29 Modo Street View (SV) en editor de Unity	44

Figura 30 Ejecución del modo Street View (SV)	45
Figura 31 Ejecución del modo Street View (SV)	45
Figura 32 Modelo con textura de 4096 pixeles	46
Figura 33 Modelo con textura de 1024 pixeles	47
Figura 34 Modelo 3D v9 (final) realizado en colaboración tesistas de Diseño Gráfico	47
Figura 35 Estructura final del script de gestión de UI en el Inspector de Unity	48
Figura 36 Mensajes en consola del proceso de encriptación y desencriptación.	48
Figura 37 Configuración de Addressables	49

CAPÍTULO I

INTRODUCCIÓN

1.1 Tema

Para la redacción de este componente guíese en el Plan de Trabajo de Integración Curricular aprobado en su momento por Honorable Consejo Directivo de la facultad.

1.2 Problema de investigación

Hoy en día, muchas personas enfrentan dificultades para explorar lugares o asistir a eventos por falta de tiempo, accesibilidad o información sobre el espacio. Para quienes deben conocer entornos extensos o complejos, como un campus universitario o un centro de convenciones, esta falta de orientación puede resultar frustrante e intimidante, afectando el aprovechamiento del lugar y aumentando el tiempo necesario para moverse con confianza. En este contexto, los paseos virtuales ofrecen una solución accesible al permitir que las personas exploren ubicaciones clave de forma remota, familiarizándose de antemano con el entorno. Esta experiencia interactiva facilita una navegación más eficiente y reduce el estrés, ofreciendo una forma inclusiva de acceso a la información para todo tipo de usuarios. En la Universidad Técnica del Norte (UTN) específicamente en su campus de el Olivo hay alrededor de 12.000 personas entre docentes, estudiantes y funcionarios en la jornada diurna y nocturna (Universidad Técnica del Norte, n.d.). Al haber tantos puntos de referencia y lugares a los cuales ir, los estudiantes tienen problemas para ubicarse y entender cómo se estructura el campus. A pesar de que la universidad realice tours e instale mapas en ciertos puntos estratégicos, para algunos estudiantes puede llegar a ser confuso o insuficiente como se establece en la **Figura 1**. La comunidad estudiantil se va actualizando de acorde al número de egresados, ingresados y reintegrados conforme pasa el tiempo, el cambio de la infraestructura y administración en este lapso genera la necesidad de plantear formas óptimas para la fácil orientación y ubicación dentro de este campus, y a su vez que este medio sea actualizable.

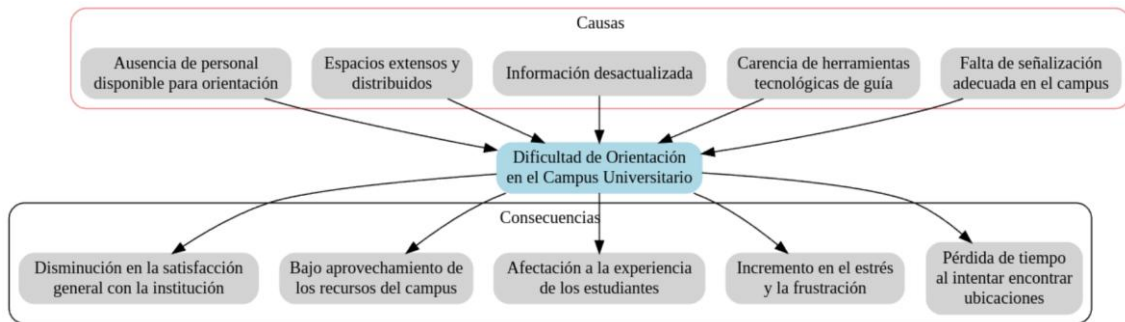


Figura 1. Árbol de problemas

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un tour virtual “Paseo UTN” con Realidad Aumentada para la Universidad Técnica del Norte utilizando el marco de trabajo SCRUM y la ISO/IEC 25022.

1.3.2 Objetivos Específicos

- Redactar un marco teórico sobre el uso de Unity con todas las herramientas y tecnologías a usar
- Desarrollar el tour virtual “Paseo UTN” para la Universidad Técnica del Norte usando la metodología SCRUM
- Evaluar calidad del uso de la aplicación usando la ISO/IEC 25022

1.4 Alcance

El presente proyecto pretende desarrollar un aplicativo móvil denominado “Paseo UTN” para facilitar la orientación y guía a la comunidad universitaria de la UTN, y personas en general con el fin de visitar virtualmente el campus El Olivo. El aplicativo será desarrollado con el motor gráfico Unity centrándose en entornos 3D para exteriores, aplicando la tecnología de realidad aumentada y el uso de los paquetes de ARCore. Además, se desarrollará un subsistema que replique el modo Street View de Google Maps para interiores. Se realizará un análisis de las diversas técnicas de modelado en 3D para el campus El Olivo, con el fin de determinar cuál es la técnica que permitirá obtener una mayor fidelidad a la realidad sin sacrificar demasiado rendimiento. Este análisis se realizará debido a que es un aplicativo dirigido principalmente a dispositivos móviles. El aplicativo “Paseo UTN” desarrollado en Unity requerirá de acceso a una base de datos

para brindar información actualizable que se podrá modificar mediante una aplicación web como se contempla en la **Figura 2**.

El desarrollo de la estructura de la base de datos, la API y la aplicación web están contempladas como parte de el mismo proyecto. El aplicativo móvil “Paseo UTN” se desarrollará como una aplicación individual que podrá vincularse con el aplicativo “UTN móvil” como acceso directo a modo de launcher.

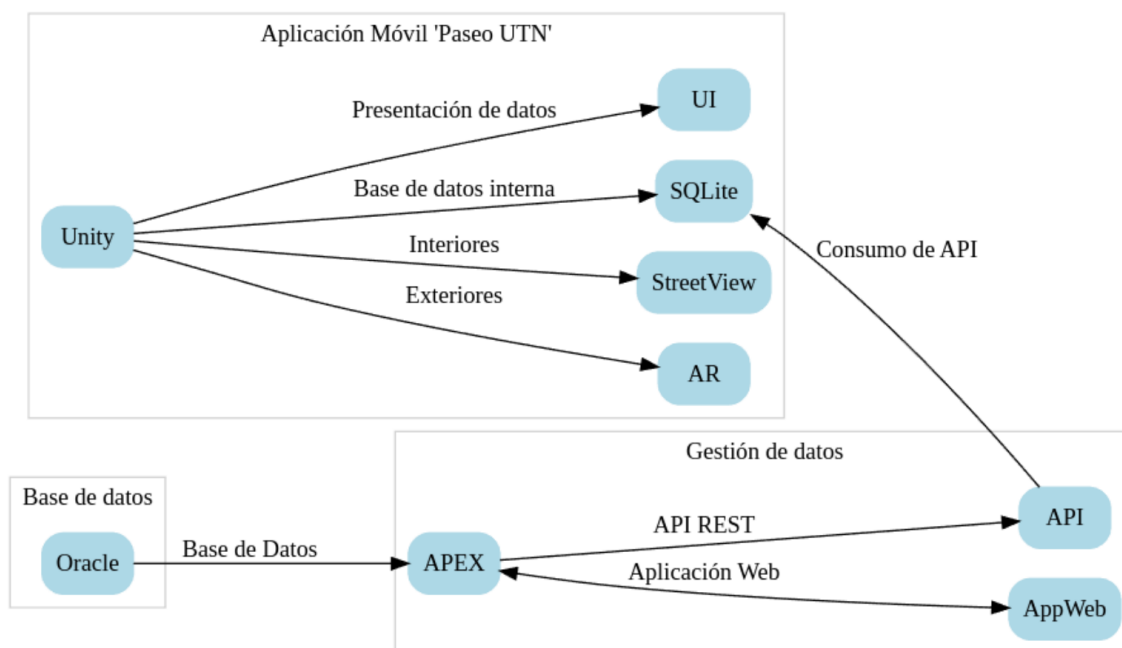


Figura 2 Relación de componentes

1.5 Metodología

El presente trabajo de investigación es de tipo aplicada, ya que busca desarrollar una aplicación móvil llamada "Paseo UTN" para facilitar la orientación en el campus El Olivo, proyectando su posible integración en el futuro aplicativo “Móvil UTN” de la UTN. Las actividades relacionadas a cada objetivo se desarrollarán de la siguiente manera:

1. **Investigación Documental:** Se llevará a cabo una investigación documental sobre técnicas de modelado en 3D y tecnologías de realidad aumentada. Se consultarán fuentes científicas accesibles mediante las bases de datos institucionales de la UTN. Para gestionar las referencias, se usará el gestor bibliográfico Mendeley.
2. **Metodología Ágil (SCRUM):** La metodología ágil SCRUM será aplicada para organizar el desarrollo en tres fases principales:
 - a. **Pre-Juego:** Se definirán los requisitos y funcionalidades del sistema, tales como el modo de exploración en 3D y los elementos interactivos de

- AR. También se seleccionarán las tecnologías y herramientas a utilizar, incluyendo Unity para el desarrollo, técnicas de modelado 3D y AR.
- b. **Juego:** Durante esta fase, se desarrollarán los siguientes componentes:
 - i. El modelo 3D del campus El Olivo, dividido en edificios y áreas exteriores.
 - ii. La integración de realidad aumentada, que permitirá a los usuarios explorar los exteriores en RA.
 - iii. El subsistema de visualización en modo "Street View", permitiendo la navegación en interiores.
 - iv. La conexión con la base de datos para obtener información actualizable de los edificios y puntos importantes.
 - v. Aplicativo web para modificar la información insertada en la base de datos.
 - c. **Post-Juego:** Se llevarán a cabo pruebas de funcionalidad y aceptación con estudiantes para asegurar que la aplicación cumple con los requisitos y ofrece una experiencia de usuario adecuada
3. **Evaluación de la Aplicación:** Una vez concluido el desarrollo, la aplicación móvil será evaluada bajo la norma ISO/IEC 25022 para analizar su calidad en uso. La evaluación se llevará a cabo mediante encuestas y pruebas prácticas con estudiantes, midiendo aspectos como la usabilidad, eficiencia y satisfacción en la orientación y navegación dentro del campus.

1.6 Justificación

La presente investigación está alineada con el objetivo 4 de los Objetivos de Desarrollo Sostenible (ODS), que busca garantizar una educación inclusiva, equitativa y de calidad, y promover oportunidades de aprendizaje durante toda la vida para todos (Organización de las Naciones Unidas, 2015). Además, los lineamientos territoriales de la primera directriz en el Plan de Creación de Oportunidades (2021-2025) enfatizan en el fortalecimiento de las TIC y sistemas de información relacionados con el desarrollo humano (Secretaría Nacional de Planificación, 2021).

Justificación Tecnológica. – La agenda de Transformación Digital del Ecuador 2022- 2025 realizada por el Ministerio de Telecomunicaciones y Sociedad de la Información. Específicamente en su punto 7.1 - Pilar 2 la necesidad de mejorar la infraestructura tecnológica de hardware y software con el objetivo de mejorar la calidad de los servicios públicos consiguiendo una infraestructura tecnológica publica más eficiente y robusta [1]. Bajo esta perspectiva, los estudiantes de la UTN en todas las modalidades de estudio son los principales beneficiarios del proyecto. Además, el personal administrativo y académico, así como la comunidad universitaria en general, se beneficiarán al contar con una guía virtual actualizada.

1.7 Antecedentes

Tabla 1 Antecedentes del Proyecto

Contexto: Local AR Typical-Food [2].	El Proyecto de investigación es un apoyo a la difusión de platos típicos de la provincia de Imbabura. El objetivo de la aplicación es brindar a los usuarios una experiencia tanto educativa como interactiva, centrada en la culinaria tradicional, mediante el uso de la realidad aumentada que muestra detalles sobre cada plato. El aporte del proyecto se centrará en que el uso de realidad aumentada tendrá un propósito de ayuda y guía usando la misma tecnología de realidad aumentada y lectura de códigos QR.
Contexto: Local Ibarra Adventure [3].	El proyecto nace de la idea de contribuir en la revalorización del patrimonio cultural de los monumentos icónicos de la ciudad de Ibarra, mediante el uso de nuevas tecnologías como es la Realidad Aumentada. El aporte del presente proyecto será aprovechar las tecnologías de realidad aumentada para señalar los monumentos icónicos de la universidad además de dar guía de ubicación hacia estos.
Contexto: Nacional Diseño de un recorrido virtual del campus universitario como propuesta para implementación en el portal web de la universidad nacional de Loja [4].	Este trabajo se centra en la creación de un recorrido virtual interactivo del campus universitario utilizando herramientas de realidad virtual y 3D. El aporte del proyecto será la creación de un tour dividido en la parte de

	exteriores e interiores además de la implementación de pantallas con información dinámica para un uso más práctico que demostrativo
Contexto: Nacional Tour Virtual 360 - Sede Cuenca [5].	El "Tour Virtual 360 - Sede Cuenca" es un proyecto diseñado para ofrecer una experiencia inmersiva y accesible que permite a los usuarios explorar virtualmente la sede de Cuenca de una institución educativa, como una universidad. El aporte a este proyecto se presentará al unir ambas tecnologías de realidad aumentada y modo 360° denominado Street View por parte del presente proyecto.
Contexto: Internacional Virtual Tour – Harvard [6].	El Tour Virtual de Harvard ofrece una experiencia interactiva que permite a los usuarios explorar el campus de la universidad desde cualquier lugar. A través de recorridos en 360 grados, los visitantes pueden conocer los edificios emblemáticos, áreas comunes y espacios académicos. El tour incluye narraciones de estudiantes y datos sobre la vida universitaria, brindando una visión completa de lo que significa estudiar en Harvard. El aporte del presente proyecto será la adición de tecnologías de Realidad Aumentada y la posible adición de narrativas a los monumentos icónicos de la UTN.
Contexto: Internacional	El tour virtual de la Universidad de Oklahoma (OU) ofrece una experiencia

Virtual Tour – University of Oklahoma [7].	inmersiva y accesible para quienes desean explorar el campus sin estar físicamente presentes. El aporte del presente proyecto será la adición de tecnologías de Realidad Aumentada con el fin de dar un toque inmersivo en las estructuras de UTN.
--	--

CAPÍTULO II

MARCO TEÓRICO

2.1 Conceptos Fundamentales

2.1.1 Realidad Aumentada (RA).

2.1.1.1 Definición de realidad aumentada.

La Realidad Aumentada (RA) es una tecnología que permite la combinación del entorno físico y real con información del entorno virtual, esto con el fin de modificar la percepción física del usuario [8]. Mediante dispositivos como smartphones, tabletas o gafas especializadas se consigue un nivel de interactividad y obtención de información superior en comparación con los métodos tradicionales.

El concepto de realidad aumentada está relacionado precisamente con esta última característica, es decir, en cómo la tecnología puede ayudarnos a enriquecer nuestra percepción de la realidad [9].

2.1.1.2 Elementos básicos de realidad aumentada.

- **Capturador de imágenes de la realidad:** Este elemento puede ser la cámara de cualquier dispositivo sin necesidad de limitarse a cámaras de smartphones, aunque esta sea la opción más común. La RA se puede aplicar con cualquier cámara que tenga la resolución suficiente para que el software pueda interpretar las imágenes [9].

- **Proyector de imagen sintetizada:** Es un medio por el cual se podrá observar la superposición de objetos y/o imágenes para obtener una percepción diferente de la realidad [9].
- **Procesador:** Su cometido es el de interpretar la información del mundo real que recibe el usuario, generar la información virtual y mezclarla de forma adecuada en pantalla, en este contexto el software es el procesador [9].
- **Activador de realidad aumentada:** La complejidad de la técnica requiere dispositivos con una serie de componentes como sensores integrados (acelerómetro, brújula, GPS, etc.) y marcadores externos (códigos QR, imágenes, objetos tangibles, etc.) que permiten identificar la posición y orientación [9].

Aunque en un mundo ideal la RA se podría llevar a cabo eliminando varios de los elementos mencionados anteriormente evitando la necesidad de procesar imágenes o proyectar las imágenes en una pantalla externa, actualmente dichos elementos son los elementos básicos para usar la tecnología de RA [9].

2.1.1.3 Niveles de realidad aumentada.

- **Nivel 0:** Hace uso de códigos QR como hiperenlaces para conectar el mundo real con un sitio web. Aunque este nivel no cumple estrictamente con la definición de RA, se le considera una etapa introductoria por su capacidad de vincular elementos del entorno real con información digital [10].
- **Nivel 1:** Usa la tecnología de RA mediante marcadores bidimensionales usados como activadores permitiendo vincular el entorno real con el virtual. Estos son códigos QR o patrones específicos y generalmente se superponen modelos 3D encima de estos [10].



Figura 3 Ejemplo de RA con imágenes [8].

- **Nivel 2:** Este nivel no hace uso de marcadores. Los activadores son imágenes, objetos o localizaciones GPS que, mediante el uso de los sensores integrados en el dispositivo permiten localizar y superponer una capa de información sobre puntos de interés del entorno [10].



Figura 4 Realidad aumentada con geolocalización [10].

- **Nivel 3:** También conocido como “Visión Aumentada”, incorpora la RA en gafas o dispositivos que se integran directamente con el usuario sin la necesidad de usar las manos. Existen pocos proyectos que logran este nivel de RA siendo como se

muestra en la figura 3: Google Glass, Apple Vision Pro y Meta Quest 3 las más conocidas actualmente [10].



Figura 5 Realidad aumentada con Google Glasses [10].

2.1.2 Modelado 3D

2.1.2.1 Fundamentos básicos del modelado 3D.

El modelado 3D es el arte de crear objetos tridimensionales digitalmente mediante la unión de formas básicas como vértices, bordes y directamente caras que también son conocidas como polígonos. Hoy en día, cualquier persona puede iniciar en el mundo del modelado 3D como si de un arte como la pintura o la música se tratase.

“Todos los elementos que forman una escena 3D las conforman polígonos.” [11].

2.1.2.2 Herramientas de creación de modelos 3D.

Para la creación de modelos 3D existen varias opciones de software especializado, si se buscase el mejor software para modelado 3D basándose en 3 características claves; utilizar poco espacio de instalación, bajo consumo de recursos del computador y ser de software libre. La mejor opción sería Blender puesto que además de cumplir con las 3 características anteriores este es un programa dedicado especialmente para el modelado, iluminación, renderizado, animación y muchas otras características de uso profesional [12].

2.1.2.3 Modelos 3D optimizados para aplicaciones móviles.

“Con menor número de polígonos, la optimización es más rápida” [11].

El nivel de detalle que se le da a un modelo 3D influye directamente en el rendimiento que tendrá la renderización de este. Por esto en distintos escenarios es de suma importancia manejar diferentes LOD (Level Of Detail) comparándolos en la ilustración 4. Además de pensar únicamente en la capacidad del hardware es importante

entender que la calidad visual del producto podrá no variar y aumentar su rendimiento si se aplican técnicas de optimización de la forma adecuada; la reducción de LOD al renderizar objetos a la distancia, deshabilitar la renderización de otros objetos que no estén siendo visualizados por la cámara, etc [13].

Comparación del impacto visual y mediciones					
Modelo					
Vértices	~5500	~2880	~1580	~670	140
Notas	Máximo detalle, para primeros planos.				Detalle mínimo, objetos muy lejanos.

Figura 6 Ejemplo de niveles de detalle (LOD) [13].

El modelado 3D se puede dividir en 2 criterios principales dependiendo del objetivo y destino que tenga el modelo.

- A. Modelado de objetos para Videojuegos (sencillos).
- B. Modelado para producción cinematográfica o animación.

Los modelos creados para un tour virtual se modelarían como si del contexto de un videojuego se tratase pues en este caso los modelos están optimizados para mostrarse en pantalla sin secuencias pre-procesadas como se muestra en la ilustración 5, ni almacenadas previamente y es de suma importancia que su visualización pueda cambiar tantas veces por segundo sean necesarias para que la experiencia del usuario sea fluida y responsiva [11].

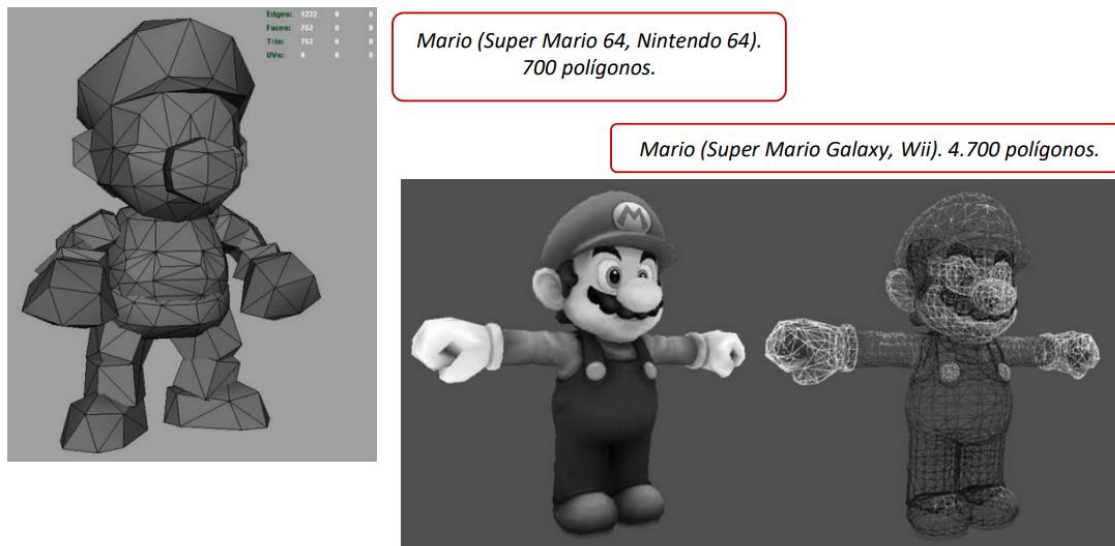


Figura 7 Comparativa de cantidad de polígonos en videojuegos distintos [11].

2.1.3 Tours Virtuales.

Un tour virtual es una experiencia que permite visitar diferentes lugares sin la necesidad de estar presencialmente integrando panoramas interactivos con medios como fotos, videos, audios y mapas. Su principal diferencia con presentaciones simples como diapositivas y/o videos es la interactividad pues un tour virtual tiene como objetivo que el usuario tenga libertad de navegación por los panoramas [14], [15].

Los tours virtuales se basan en convertir imágenes panorámicas en formatos esféricos inmersivos donde el usuario pueda explorar de una forma detallada y navegable mediante sus dispositivos como computador, teléfono, TV, etc [14].

Cada tour virtual puede tener características como [14]:

- Interactividad con hotspots.
- Reproducción de sonidos según la orientación.
- Efectos visuales.
- Conexión entre varios panoramas.
- Información adicional de ubicaciones
- Rutas prediseñadas
- Piloto guía automático alternable

Generalmente un tour virtual sigue 3 pasos principales [15]:

1. **Fotografía 360°:** Captura de imágenes panorámicas con cámaras especializadas.

2. **Procesamiento y edición:** Ensamblado de imágenes con software para crear una vista continua, añadiendo elementos interactivos.
3. **Integración y visualización:** Publicación del tour en una plataforma digital para el acceso y uso de los usuarios.

2.2 Tecnologías y Herramientas.

2.2.1 Unity y Realidad aumentada.

2.2.1.1 ¿Qué es Unity?

Unity es un motor de desarrollo de videojuegos en 2D y 3D de licencia gratuita (bajo ciertas condiciones) que ha ganado bastante popularidad debido a su facilidad de uso, el bajo peso que tiene en comparación con otros motores de desarrollo y la extensa cantidad de herramientas que dispone a los usuarios [16].

Para funcionar, Unity hace uso de escenas (scenes) y activos (assets):

- Escenas: Son el espacio en donde se agregan los activos y todo el contenido que el desarrollador quiere incluir en su proyecto [17].

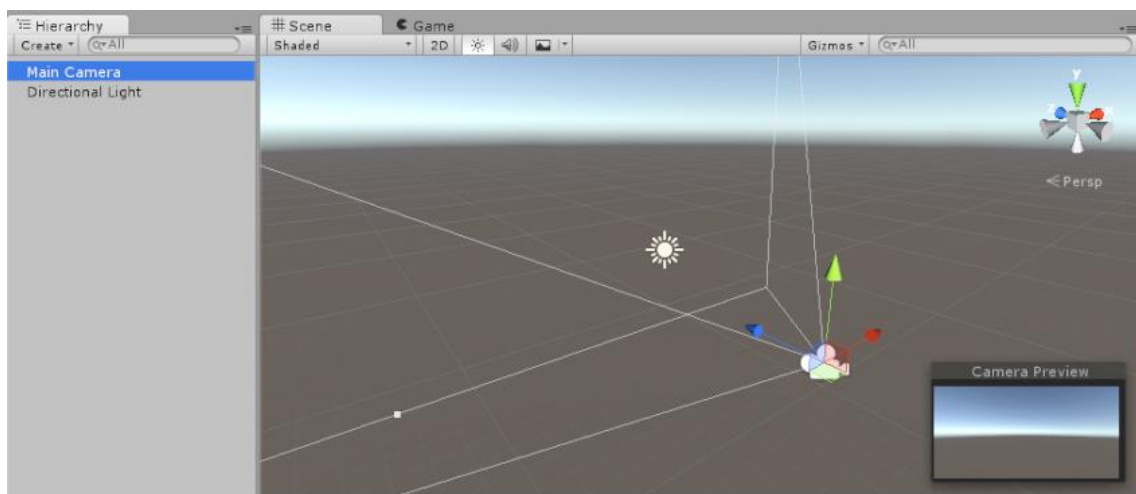


Figura 8 Escena en el editor de Unity [17].

- Activo: Es cualquier objeto que se desee importar en el proyecto. Archivos de cualquier formato como: imágenes, audios, UI, modelos 3D, fuentes de texto, etc [18] es necesario destacar que una vez agregado en escena, un activo pasa a ser parte de un GameObject.

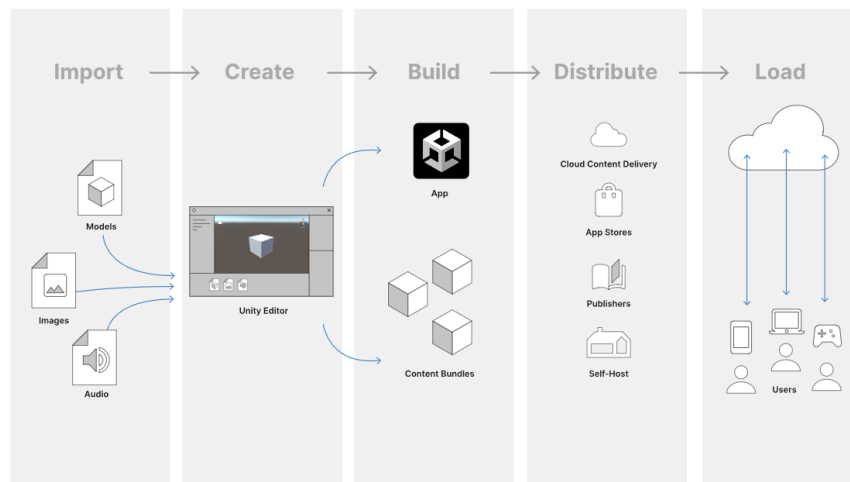


Figura 9 Flujo de trabajo de assets en Unity 3D [18].

2.2.1.2 Realidad Aumentada en Unity 3D.

Para lograr el efecto que se espera encontrar de la realidad aumentada con Unity 3D superpone activos en la cámara del dispositivo en el mundo real del usuario determinando su posición mediante los sensores de este buscando marcadores o seleccionando puntos como marcadores siendo estos superficies planas, objetos u otras ubicaciones convenientes [19], teniendo en cuenta que en el tour virtual la universidad podrá mostrarse tanto con o sin marcadores centrándose en distintos enfoques, al leer un código QR específico se podrá mostrar una edificación y al no leer ningún código QR se mostraría la universidad completa.

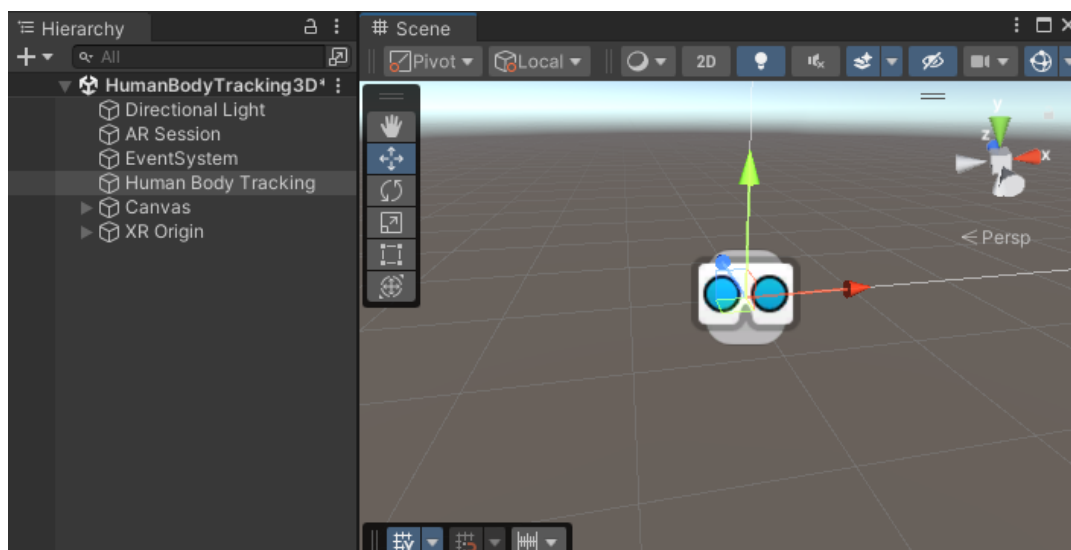


Figura 10 Escena de Realidad Aumentada en el Editor de Unity [19].

2.2.2 Blender.

Es una herramienta multiplataforma gratuita y de código abierto bastante ligera desarrollada con el fin de permitir a cualquier usuario modelar objetos en 3D [20].

2.2.2.1 Características principales de Blender.

- Es una herramienta completa para crear contenido 3D, con funciones como modelado, renderizado, animación, edición de video, efectos visuales y simulaciones.
- Es multiplataforma por lo que funciona en distintos sistemas operativos.
- Tiene una interfaz personalizable y funciona rápido y eficientemente.
- No necesita instalación tradicional y puede ejecutarse desde cualquier carpeta.
- Cuenta con una comunidad activa para soporte.



Figura 11 Imagen renderizada y procesada con Blender [20].

Blender no solo es útil para modelar muebles, sino también para crear escenarios completos que pueden ser utilizados en experiencias de realidad aumentada (AR). Esto sugiere que Blender puede ser una herramienta efectiva para visualizar proyectos arquitectónicos en entornos interactivos [21].

2.2.3 Metodología SCRUM.

Scrum es un marco de trabajo ágil, se recomienda usar para proyectos nuevos y con alta posibilidad de cambio que permitan a las personas, equipos y organizaciones llevar a cabo soluciones bastante adaptativas a trabajos complejos. Aunque teóricamente

está incompleto, SCRUM busca guiar con las relaciones e interacciones de los grupos antes que directamente dar instrucciones detalladas [22].

2.2.3.1 Conceptos básicos de SCRUM.

- SCRUM Team: Es el equipo multifuncional de alrededor de 10 personas que está a cargo de todas las actividades relacionadas con el producto, está conformado por tres responsabilidades específicas: SCRUM Master, Product Owner y Developers [22].
- Sprint: Es el evento más importante de SCRUM pues aquí se llevan a cabo los objetivos planteados en el Product Backlog, tiene una duración constante de un mes o menos. Una vez terminado un Sprint se lleva a cabo una Sprint Review para inspeccionar los resultados del Sprint y determinar futuras adaptaciones para el siguiente sprint que se iniciará una vez terminado el Sprint previo [22].
- Developers: Son las personas encargadas de hacer un avance que represente un valor para el producto, son la mano de obra directa del proyecto y sus habilidades deberán ser útiles para el desarrollo del producto [22].
- Product Owner: Es el líder encargado de gestionar los objetivos de cada Sprint, plantear de forma clara y transparente el Product Backlog y delegar las responsabilidades a los Developers. Es necesario destacar que el Product Owner es responsable de que el trabajo se realice [22].
- SCRUM Master: Es la persona encargada de que el SCRUM Team entienda la teoría de SCRUM y logre que este se vuelva efectivo, estos apoyan al equipo mejorando sus prácticas liderando y capacitando. Además ayuda al Product Owner a gestionar el Product Backlog [22].

2.2.3.2 SCRUM para el desarrollo en Unity.

Cuando se trata de desarrollar videojuegos se tiende a usar metodologías ágiles gracias a que esta permite hacer un desarrollo en base a lo empírico y adaptarse fácilmente a los cambiantes requerimientos [23].

Como se estableció anteriormente, el desarrollo de un tour virtual se tratará como si el desarrollo de un videojuego se tratase por el uso que se le dará y por la naturaleza del motor de desarrollo en el que se llevará el proyecto.

2.2.4 ISO/IEC 25022

2.2.4.1. Familia de normas ISO/IEC 25000

[24] ISO/IEC 25000, conocida como SQuaRE, es una familia de normas que establece un marco común para evaluar la calidad del software. Se basa en la evolución de las normas ISO/IEC 9126 (modelo de calidad del software) e ISO/IEC 14598 (evaluación del software). Está compuesta por cinco divisiones:

1. **Gestión de Calidad (ISO/IEC 2500n):** Define modelos, términos y directrices generales, incluyendo la arquitectura SQuaRE (ISO/IEC 25000) y la gestión de evaluación (ISO/IEC 25001).
2. **Modelo de Calidad (ISO/IEC 2501n):** Presenta modelos detallados para evaluar la calidad interna, externa y en uso del software, destacando ISO/IEC 25010 (modelo de calidad del software) e ISO/IEC 25012 (calidad de datos).
3. **Medición de Calidad (ISO/IEC 2502n):** Proporciona métricas y guías para evaluar la calidad del software y los datos, incluyendo ISO/IEC 25020 (modelo de referencia), ISO/IEC 25021 (métricas base), ISO/IEC 25022 (calidad en uso), ISO/IEC 25023 (calidad del software) e ISO/IEC 25024 (calidad de datos) [24].

2.2.4.2. Resumen

[25] ISO/IEC 25022:2016 define medidas de calidad en uso basadas en las características de ISO/IEC 25010. Proporciona un conjunto básico de métricas y explica cómo evaluar la calidad en uso de sistemas y productos de software en contextos reales.

Incluye anexos informativos sobre la medición de la cobertura del contexto, normalización de medidas, relación con ISO 9241-11, evaluación de calidad en uso y conceptos de medición.

Las métricas no establecen rangos fijos, ya que estos dependen del contexto y las necesidades del usuario. Se aplican a cualquier sistema humano-computador y ayudan a gestionar el desarrollo, adquisición, evaluación y mantenimiento del software [25].

2.3 Trabajos Relacionados

2.5.1 Virtual Campus Tour Application through Markerless Augmented Reality Approach.

[26] desarrolló una aplicación de realidad aumentada AR-UTHM Tour para permitir a los usuarios explorar virtualmente el campus de la Universidad Tun Hussein Onn Malaysia (UTHM), ofreciendo una experiencia interactiva e inmersiva que facilita el conocimiento de las instalaciones y cursos sin necesidad de una visita física. Utilizando el modelo ágil de desarrollo de software, el proyecto incluyó diseño gráfico, modelado 3D en Blender, integración en Unity con scripting en C# y AR sin marcadores mediante el SDK de Wikitude. La aplicación fue probada por 30 estudiantes, mostrando un alto nivel de aceptación, con más del 50% ejecutando sesiones de AR sin errores. Sin embargo, enfrenta limitaciones como la dependencia de actualizaciones en la base de datos y la necesidad de mejorar la interactividad y la cantidad de modelos 3D.

2.5.2 Virtual Campus Tour Berbasis Web Menggunakan Teknologi Augmented Reality.

[27] desarrolló un tour virtual basado en web con realidad aumentada para simular el entorno del campus de la Universidad Muhammadiyah Magelang (UNIMMA), con el objetivo de ayudar a futuros estudiantes, especialmente aquellos provenientes de otras ciudades, a comprender mejor las instalaciones del campus. Utilizando herramientas como Unity y Blender 3D, y siguiendo el ciclo de vida de desarrollo multimedia (MDLC), el proyecto abarcó etapas de diseño y pruebas enfocadas en la creación de contenido visual y la simulación interactiva del entorno. El resultado destacado fue la creación de un tour virtual que permite explorar el campus e informar sobre sus diferentes áreas, facilitando la adaptación de nuevos estudiantes. Sin embargo, enfrenta limitaciones como la necesidad de dispositivos adecuados, conexión a Internet y una base de datos actualizada para garantizar la precisión del contenido.

2.5.3 Aplikasi Virtual Tour Islamic Center Berbasis Android.

[28] presenta el desarrollo de la aplicación "Virtual Tour Islamic Center", que ofrece recorridos interactivos en 3D y videos panorámicos para explorar el Islamic Center Tulang Bawang Barat. Utilizando el modelo MDLC, se definieron objetivos, diseñaron interfaces, recolectaron materiales, desarrollaron la aplicación en Unity 3D, se realizaron pruebas y se distribuyó el producto. Como resultado, se creó una herramienta interactiva

que mejora la experiencia del usuario y promueve el centro, aunque enfrenta limitaciones como dependencia de internet, requisitos técnicos, curva de aprendizaje y contenido limitado.

2.5.4 Memórias Tridimensionais do Grande ABC: Tour Virtual 3D pela Vila de Paranapiacaba.

El proyecto "Memórias Tridimensionais do Grande ABC", desarrollado por [29], es un entorno virtual que replica en 3D la Vila de Paranapiacaba, incorporando modelos fieles a las construcciones originales e información histórico-cultural interactiva, destacando su importancia histórica y su potencial como patrimonio de la humanidad. La metodología incluyó un equipo multidisciplinario, visitas al sitio, uso de plantas, fotografías y modelado 3D con herramientas de computación gráfica, inspirándose en simuladores de paseo. El resultado principal es un tour virtual que permite explorar la Parte Baja de la vila con interacción y contenido multimedia. Sin embargo, la precisión arquitectónica se ve limitada por el uso de plantas y fotos, y el hardware necesario puede restringir el acceso, destacando la necesidad de futuras mejoras.

2.5.5 Virtual Tour With Voice Assistant using Extended Reality.

El proyecto presentado por [30] "Virtual Tour With Voice Assistant using Extended Reality" tenía como objetivo crear un recorrido virtual de un campus universitario que incorporara un asistente de voz basado en IoT. Para lograrlo, los autores utilizaron el motor de juego Unity 3D y varios de sus componentes, incluyendo objetos 3D para construir el entorno virtual, teletransportación para la navegación y scripts para la interacción con objetos. El resultado más relevante fue la creación de una experiencia inmersiva que permite a los usuarios explorar el campus virtualmente y obtener información mediante un asistente de voz con audio pregrabado. Sin embargo, el proyecto presenta limitaciones, como la falta de avatares interactivos, la dependencia de audio pregrabado y la necesidad de hardware específico (HTC Vive Pro 2 Kit).

2.5.6 Virtual Reality Based Virtual Tour of College Using Unity 3D.

[31] presenta un sistema de "Tour Virtual de la Infraestructura Universitaria" utilizando realidad virtual, desarrollado con Unity 3D y basado en imágenes de 360 grados del departamento de informática del K.K. Wagh Institute of Engineering Education and Research. El objetivo principal es brindar una alternativa a las visitas presenciales al campus, especialmente para aquellos que viven lejos. El sistema ofrece un

recorrido virtual del departamento, incluyendo la ruta para llegar a diferentes ubicaciones, y un módulo de búsqueda para encontrar estudiantes y personal. A pesar de su innovación, el sistema presenta limitaciones como la necesidad de un visor VR, su especificidad a una sola universidad, y la vista estática del campus. Estas limitaciones abren la posibilidad para futuros trabajos que busquen desarrollar una versión accesible desde dispositivos móviles o computadoras, crear una plataforma más generalizada para diferentes universidades, e implementar una vista en tiempo real del campus.

2.5.7 Implementing Augmented reality in Tourism.

[32] presenta el desarrollo de PazinAR, una aplicación de realidad aumentada (RA) para el turismo en la ciudad de Pazin, Croacia. El objetivo principal de la aplicación es enriquecer la oferta turística de la ciudad y mejorar la experiencia de los turistas al permitirles superponer fotos antiguas de Pazin sobre la vista actual del entorno utilizando sus teléfonos inteligentes. Para el desarrollo de PazinAR, se recopilieron fotos antiguas de la ciudad y se utilizaron el software Unity y el SDK Vuforia para implementar las funciones de RA. A pesar de que la aplicación es funcional, presenta algunas limitaciones, como la presencia de errores, la falta de interacción del usuario y la disponibilidad limitada a un solo tipo de teléfono inteligente. Estas limitaciones abren posibilidades para futuros trabajos que se centren en mejorar la estabilidad de la aplicación, diseñar funciones más interactivas y ampliar su compatibilidad con otros dispositivos.

2.5.8 Exploración en Arquitectura: Un proceso basado en Blender

El proyecto de [33] analizó y creó una nueva posibilidad para evaluar la distribución del espacio arquitectónico interior mediante un modelo virtual tridimensional interactivo. Utilizando el software libre Blender, se construyó un modelo 3D a partir de planos arquitectónicos, agregando detalles como texturas, materiales y animaciones para simular el ambiente de un edificio real. El resultado principal fue un modelo del Laboratorio de Visualización Inmersiva y Sistemas Autónomos, que permite a los usuarios explorar el espacio y experimentar su distribución y funcionamiento. A pesar del éxito, se identificaron limitaciones como la falta de un sistema métrico preciso en Blender y la influencia del nivel de detalle en el rendimiento del juego interactivo, lo que abre posibilidades para futuros trabajos enfocados en mejorar la precisión de la conversión de unidades y optimizar el nivel de detalle para una mejor experiencia de usuario.

CAPÍTULO III

DESARROLLO

El presente capítulo se enfoca en el desarrollo de la aplicación “Paseo UTN” usando las herramientas especificadas.

Para el desarrollo se usará SCRUM con sus debidas fases.

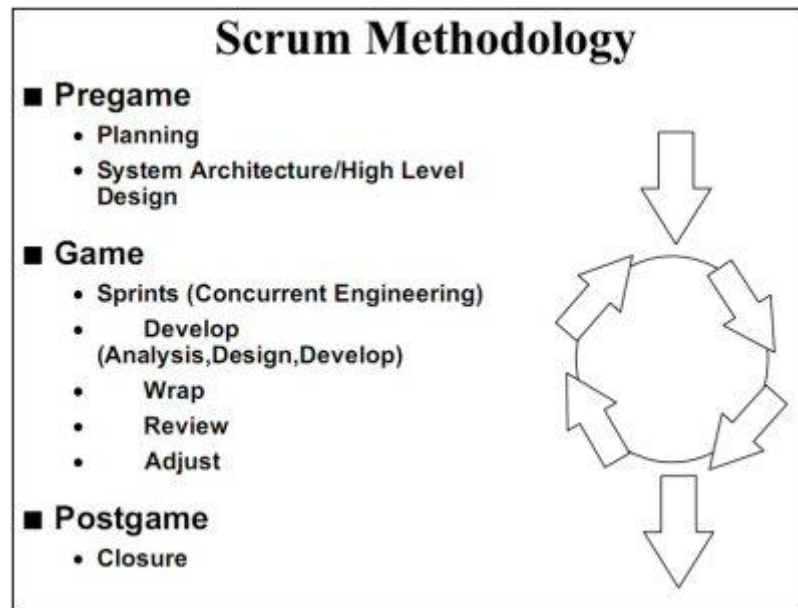


Figura 12 Fases de SCRUM [34].

3.1 Fase: Pre-juego

3.1.1 Definición de roles para el desarrollo de la aplicación:

Para el desarrollo de la aplicación se deben definir los roles establecidos para SCRUM.

Tabla 2 Roles de SCRUM

Nombre	Rol	Cargo
Cathy Guevara	Product Owner	Director del Trabajo de Grado Docente UTN
Victor Echeverría	Scrum Master Equipo de desarrollo	Estudiante Tesista

Dirección de Desarrollo Tecnológico e Informático	Stakeholder	Departamento TI de la Institución
--	--------------------	--

3.1.2 Elicitación de requisitos

Los requisitos del sistema fueron definidos mediante la elaboración de historias de usuario, tomando como base el enfoque de trabajo de SCRUM [35]. Siguiendo este enfoque, se realizó una reunión con el Product Owner, PhD. Cathy Guevara, en la cual se revisaron las funcionalidades del proyecto y se definieron las historias de usuario a desarrollar, facilitando su adecuada formulación y orden de prioridad dentro del proceso de construcción de la aplicación.

Tabla 3 Historia de Usuario 1

Historia de usuario		
ID: HU-01	Usuario: Usuario final	
Nombre:	Visualización del modelo 3D	
Prioridad: Alta	Dependencia: N/A	History Points: 5
Descripción:	Como usuario quiero visualizar un modelo 3D completo de la universidad para comprender su distribución general, su forma y la ubicación relativa de sus edificios antes de interactuar con otras funcionalidades.	
Criterios de aceptación:	El modelo 3D se carga correctamente al iniciar la aplicación.	
	El modelo representa fielmente la estructura general del campus.	
	El modelo es visible tanto en modo NoAR como AR.	
	El rendimiento es estable y no presenta caídas notorias de FPS.	

Tabla 4 Historia de Usuario 2

Historia de usuario

ID: HU-02	Usuario: Usuario final
Nombre:	Interacciones básicas del modelo
Prioridad: Alta	Dependencia: HU-01 History Points: 8
Descripción:	Como usuario quiero poder mover, rotar y acercar o alejar el modelo 3D mediante gestos táctiles para explorarlo de forma intuitiva y detallada desde diferentes ángulos.
Criterios de aceptación:	El gesto de arrastre permite mover el modelo de forma fluida.
	El gesto de pellizcar permite hacer zoom in y zoom out progresivamente.
	El gesto de rotación permite girar el modelo sobre su eje.
	La interacción responde en tiempo real.
	Los gestos no interfieren entre sí.

Tabla 5 Historia de Usuario 3

Historia de usuario	
ID: HU-03	Usuario: Usuario final
Nombre:	Guía de gestos
Prioridad: Media	Dependencia: HU-01 History Points: 3
Descripción:	Como usuario quiero visualizar una guía explicativa de gestos para entender cómo interactuar correctamente con el modelo 3D y aprovechar todas las funcionalidades de la aplicación.
Criterios de aceptación:	La guía muestra claramente cada gesto y su función.
	Incluye texto descriptivo y apoyo visual.
	Se muestra al primer inicio o desde un acceso manual.
	Puede cerrarse para no interrumpir la experiencia.

Tabla 6 Historia de Usuario 4

Historia de usuario	
----------------------------	--

ID: HU-04	Usuario: Usuario final
Nombre:	Modo NoAR
Prioridad: Alta	Dependencia: HU-01 History Points: 5
Descripción:	Como usuario quiero utilizar la aplicación en modo NoAR para visualizar el modelo 3D en un entorno virtual sin necesidad de usar la cámara del dispositivo.
Criterios de aceptación:	El modelo se muestra en una escena 3D tradicional.
	No se requiere acceso a la cámara.
	Todas las interacciones básicas están habilitadas.
	El modo funciona en dispositivos sin soporte AR.

Tabla 7 Historia de Usuario 5

Historia de usuario	
ID: HU-05	Usuario: Usuario final
Nombre:	Modo AR
Prioridad: Alta	Dependencia: HU-01 History Points: 8
Descripción:	Como usuario quiero utilizar el modo AR para ver el modelo 3D superpuesto en el entorno real captado por la cámara y tener una experiencia más inmersiva.
Criterios de aceptación:	Se activa la cámara del dispositivo correctamente.
	El modelo se ancla de forma estable al entorno.
	El usuario puede ajustar escala y posición.
	El modelo mantiene estabilidad durante el movimiento.

Tabla 8 Historia de Usuario 6

Historia de usuario	
ID: HU-06	Usuario: Usuario final
Nombre:	Alternar modos
Prioridad: Alta	Dependencia: HU-04, HU-05 History Points: 3

Descripción:	Como usuario quiero alternar fácilmente entre los modos NoAR y AR para elegir la forma de visualización que mejor se adapte a mi contexto.
Criterios de aceptación:	Existe un botón visible para alternar modos.
	El cambio de modo es rápido y sin errores.
	No se cierra la aplicación al cambiar de modo.
	Se mantiene la integridad del modelo.

Tabla 9 Historia de Usuario 7

Historia de usuario		
ID: HU-07	Usuario: Usuario final	
Nombre:	Selección de estructuras	
Prioridad: Alta	Dependencia: HU-02	History Points: 11
Descripción:	Como usuario quiero seleccionar una estructura específica del modelo 3D para poder interactuar con ella y acceder a información adicional.	
Criterios de aceptación:	La selección se realiza mediante toque.	
	La estructura seleccionada se resalta visualmente.	
	Solo una estructura puede estar activa a la vez.	
	La selección es clara y precisa.	

Tabla 10 Historia de Usuario 8

Historia de usuario		
ID: HU-08	Usuario: Usuario final	
Nombre:	Información de estructuras	
Prioridad: Media	Dependencia: HU-07	History Points: 2
Descripción:	Como usuario quiero visualizar información general de una estructura seleccionada para conocer su función, nombre y uso dentro de la universidad.	
Criterios de aceptación:	Existe un botón de información accesible.	
	Se muestra un panel con datos descriptivos.	
	El texto es legible y claro.	

La información corresponde a la estructura seleccionada.

Tabla 11 Historia de Usuario 9

Historia de usuario	
ID: HU-09	Usuario: Usuario final
Nombre:	Acceso al modo Street View
Prioridad: Alta	Dependencia: HU-07 History Points: 8
Descripción:	Como usuario quiero ingresar virtualmente al interior de un edificio seleccionado para recorrerlo mediante el modo Street View y conocer sus espacios internos.
Criterios de aceptación:	Existe un botón para ingresar al interior.
	Se realiza una transición clara exterior → interior.
	Solo se carga el edificio seleccionado.
	El cambio de modo es estable.

Tabla 12 Historia de Usuario 10

Historia de usuario	
ID: HU-10	Usuario: Usuario final
Nombre:	Navegación Street View
Prioridad: Alta	Dependencia: HU-09 History Points: 13
Descripción:	Como usuario quiero navegar dentro del edificio usando puntos de navegación para desplazarme entre distintas áreas de manera controlada y sencilla.
Criterios de aceptación:	Los puntos de navegación son visibles.
	El desplazamiento entre puntos es fluido.
	La cámara se posiciona correctamente.
	No se producen movimientos bruscos.

Tabla 13 Historia de Usuario 11

Historia de usuario	
ID: HU-11	Usuario: Usuario final
Nombre:	Marcadores de lugares

Prioridad: Media	Dependencia: HU-10	History Points: 5
Descripción:	Como usuario quiero visualizar carteles informativos dentro del modo Street View para identificar departamentos, aulas y otros espacios relevantes.	
Criterios de aceptación:	Los carteles aparecen en puntos clave.	
	El texto es legible a distancia razonable.	
	Cada cartel corresponde a un espacio real.	
	No obstruyen la navegación.	

Tabla 14 Historia de Usuario 12

Historia de usuario		
ID: HU-12	Usuario: Usuario final	
Nombre:	Botón de submenú	
Prioridad: Media	Dependencia: HU-01	History Points: 2
Descripción:	Como usuario quiero acceder a un submenú de opciones para realizar acciones adicionales sin salir de la experiencia principal.	
Criterios de aceptación:	El submenú es accesible desde un botón visible.	
	Se despliega y oculta correctamente.	
	No interfiere con el modelo 3D.	
	Mantiene coherencia visual.	

Tabla 15 Historia de Usuario 13

Historia de usuario		
ID: HU-13	Usuario: Usuario final	
Nombre:	Salir de la aplicación	
Prioridad: Baja	Dependencia: HU-12	History Points: 1
Descripción:	Como usuario quiero cerrar la aplicación desde el submenú para salir de forma controlada y sin errores.	
Criterios de aceptación:	La opción “Salir” está claramente identificada.	
	La aplicación se cierra correctamente.	
	No se producen bloqueos ni errores.	

Tabla 16 Historia de Usuario 14

Historia de usuario		
ID: HU-14	Usuario: Usuario final	
Nombre:	Reiniciar escena	
Prioridad: Media	Dependencia: HU-12	History Points: 2
Descripción:	Como usuario quiero reiniciar la escena de la aplicación para reubicar la cámara y el modelo 3D cuando la visualización no sea adecuada.	
Criterios de aceptación:	Se restablece la posición inicial del modelo.	
	Se reinicia la cámara correctamente.	
	Se limpian estados previos.	
	La app continúa funcionando sin reiniciarse por completo.	

Tabla 17 Historia de Usuario 15

Historia de usuario		
ID: HU-15	Usuario: Usuario final	
Nombre:	Captura de pantalla en AR	
Prioridad: Media	Dependencia: HU-05	History Points: 5
Descripción:	Como usuario quiero capturar una imagen de la experiencia en modo AR para compartirla en redes sociales u otras aplicaciones.	
Criterios de aceptación:	El botón de captura solo aparece en modo AR.	
	La imagen incluye el modelo y el entorno real.	
	Se abre el menú nativo de compartir del dispositivo.	
	La imagen se guarda correctamente.	

Tabla 18 Historia de Usuario 16

Historia de usuario		
ID: HU-16	Usuario: Sistema	
Nombre:	Gestión de estados por modo	

Prioridad: Alta	Dependencia: HU-04, HU-05, HU-09 History Points: 13
Descripción:	Como sistema quiero gestionar correctamente los estados y transiciones entre los modos NoAR, AR y Street View para garantizar estabilidad, rendimiento y ausencia de conflictos.
Criterios de aceptación:	Solo un modo puede estar activo a la vez.
	Se liberan correctamente los recursos al cambiar de modo.
	No se producen errores de cámara o renderizado.
	Las transiciones son consistentes y estables.

Tabla 19 Historia de Usuario 17

Historia de usuario	
ID: HU-17	Usuario: Sistema
Nombre:	Gestión de recursos con Unity Addressables para optimización de tamaño
Prioridad: Alta	Dependencia: HU-01, HU-09, HU-16 History Points: 13
Descripción:	Como sistema quiero utilizar Unity Addressables para cargar y descargar modelos, escenas y recursos bajo demanda con el fin de mantener el tamaño de la aplicación por debajo de los 200 MB exigidos por Google Play sin afectar la experiencia del usuario.
Criterios de aceptación:	Los assets se liberan correctamente al salir de un modo o edificio.
	El tamaño final del AAB/APK es inferior a 200 MB.
	La descarga de contenidos no bloquea la interfaz de usuario.
	La aplicación maneja correctamente errores de red o descargas incompletas.

Tabla 20 Historia de Usuario 18

Historia de usuario	
ID: HU-18	Usuario: Sistema
Nombre:	Cifrado y empaquetado seguro del modelo 3D
Prioridad: Alta	Dependencia: HU-01, HU- History Points: 13 17
Descripción:	Como sistema quiero empaquetar el modelo 3D en un formato distribuible y cifrarlo antes de incluirlo en la aplicación para proteger el contenido y evitar el acceso directo no autorizado al modelo.
Criterios de aceptación:	El modelo 3D se empaqueta correctamente en un recurso distribuible antes de integrarse a la app.
	El archivo resultante queda cifrado y no es usable/visible directamente fuera de la aplicación.
	El proceso de empaquetado y cifrado es repetible (genera el mismo resultado esperado al ejecutarlo nuevamente).
	La aplicación incluye el archivo cifrado sin romper el flujo de carga normal.
	Si el archivo no existe o está corrupto, se detecta el error y se evita un fallo de la app (mensaje o manejo controlado).

Tabla 21 Historia de Usuario 19

Historia de usuario	
ID: HU-19	Usuario: Sistema
Nombre:	Descifrado y carga segura del modelo 3D en tiempo de ejecución
Prioridad: Alta	Dependencia: HU-018 History Points: 13
Descripción:	Como sistema quiero descifrar el modelo 3D durante la ejecución de la aplicación y cargarlo correctamente en

Criterios de aceptación:	escena para permitir su uso sin exponer los archivos originales ni afectar la visualización.
	En ejecución, la app localiza el recurso cifrado y lo descifra correctamente.
	El modelo se carga e instancia en escena de forma correcta y utilizable.
	Los materiales y texturas se muestran correctamente luego de la carga (sin “rosado”, sin pérdidas evidentes).
	Si el descifrado falla (clave/archivo inválido), la app lo maneja sin cerrarse inesperadamente.
	El usuario puede continuar usando la aplicación tras la carga (no se queda bloqueada permanentemente).

3.1.3 Product Backlog

El product backlog se utilizará para **organizar, priorizar y planificar** todas las historias de usuario del proyecto *Paseo UTN*, permitiendo gestionar el desarrollo de forma incremental mediante SCRUM, asegurando que las funcionalidades de mayor valor y riesgo se implementen primero.

Prioridad	ID	Nombre de la Historia	Estimación (Puntos de Historia)
Alta	HU-01	Visualización del modelo 3D de la universidad	5
Alta	HU-02	Interacciones básicas del modelo 3D	8
Alta	HU-04	Modo NoAR	5
Alta	HU-05	Modo AR	8
Alta	HU-06	Alternar entre modos NoAR y AR	3
Alta	HU-07	Selección de estructuras	3
Alta	HU-09	Acceso al modo Street View	8
Alta	HU-10	Navegación mediante puntos en Street View	13

Alta	HU-16	Gestión de estados y modos de la aplicación	13
Alta	HU-17	Gestión de recursos con Unity Addressables para optimización de tamaño	13
Alta	HU-18	Cifrado y empaquetado seguro del modelo 3D	13
Alta	HU-19	Descifrado y carga segura del modelo 3D en tiempo de ejecución	13
Media	HU-03	Guía de gestos de interacción	3
Media	HU-08	Información general de estructuras	2
Media	HU-11	Carteles informativos en Street View	5
Media	HU-12	Submenú de opciones	2
Media	HU-14	Reiniciar escena	2
Media	HU-15	Captura de pantalla en modo AR	5
Baja	HU-13	Salir de la aplicación	1

3.2 Fase: Juego

La fase de juego corresponde a la etapa central del desarrollo del proyecto *Paseo UTN*, en la cual se implementaron de manera incremental las funcionalidades definidas previamente en el product backlog. Esta fase se desarrolló siguiendo el marco de trabajo SCRUM, dividiendo el trabajo en varios sprints, cada uno con objetivos claros y entregables funcionales.

Durante esta fase se priorizó la construcción progresiva del sistema, permitiendo validar continuamente las funcionalidades implementadas mediante pruebas visuales y de uso, así como realizar ajustes conforme avanzaba el desarrollo.

3.2.1 Sprint 1 - Visualización base y modos de interacción

El primer sprint tuvo como objetivo establecer la base visual y funcional de la aplicación, permitiendo la correcta visualización del modelo 3D del campus de la UTN y la implementación de las interacciones iniciales con el usuario.

Para la construcción del modelo 3D, se realizó un proceso de levantamiento visual de los edificios de la universidad mediante el uso de un dron, con el fin de obtener referencias reales y precisas. A partir de este material, los edificios fueron modelados de forma individual, priorizando la separación de cada estructura para facilitar su posterior interacción dentro de la aplicación. Este trabajo se desarrolló de manera colaborativa con dos estudiantes tesistas de la carrera de Diseño Gráfico, quienes aportaron en la elaboración y refinamiento visual de los modelos.

Durante este proceso se evaluó el uso de técnicas de fotogrametría; sin embargo, los resultados obtenidos no fueron adecuados para los objetivos del proyecto, ya que no permitían un control apropiado de las estructuras ni una optimización eficiente del modelo. Por este motivo, se optó por un modelado manual que ofreciera mayor control, mejor rendimiento y una integración más adecuada dentro de la aplicación.



Figura 13 Modelo 3D de la UTN realizado con Fotogrametría (15 millones de polígonos)

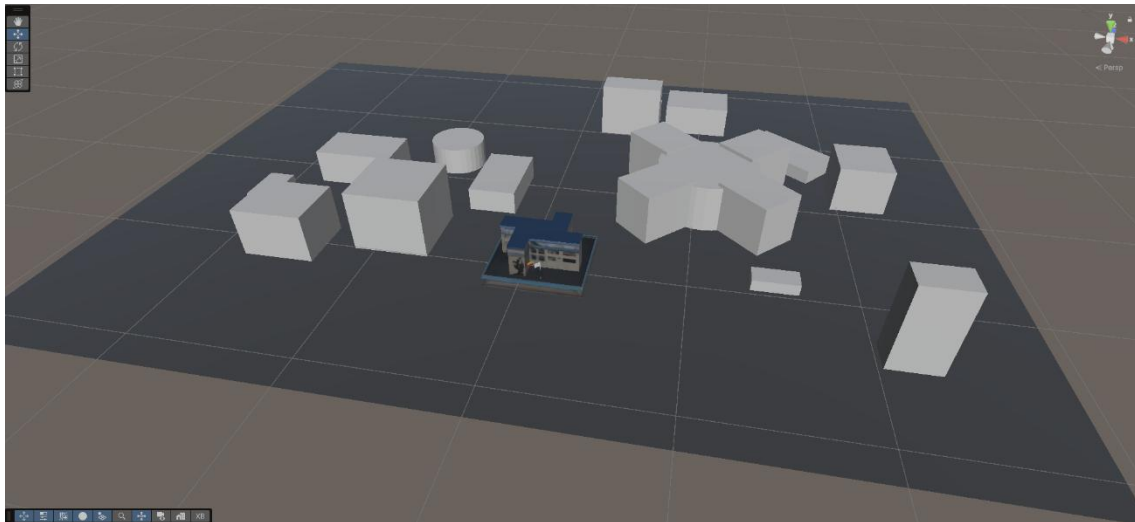


Figura 14 Modelo 3D de la UTN placeholder

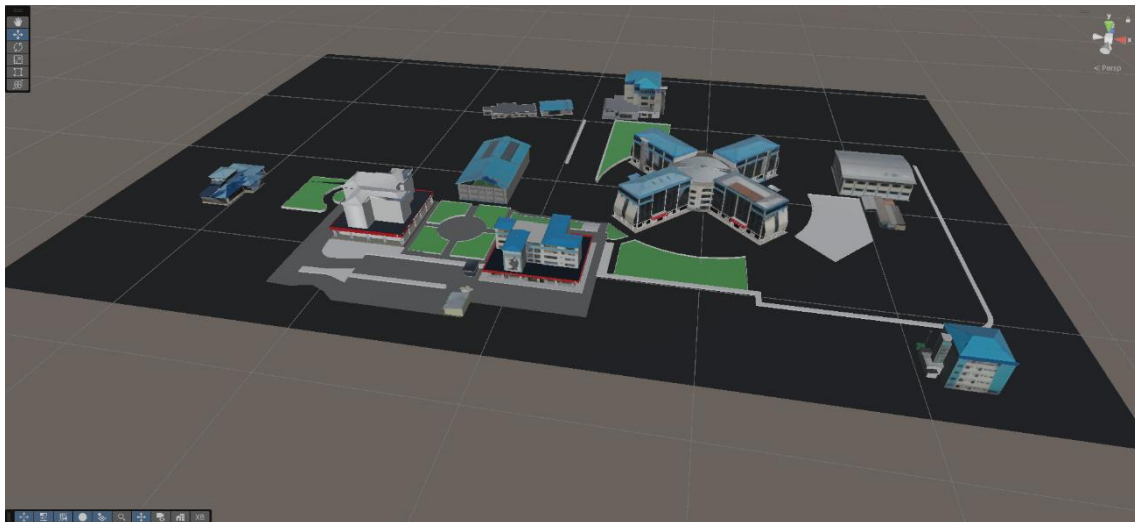


Figura 15 Modelo 3D UTN v1 realizado en colaboración tesistas de Diseño Gráfico

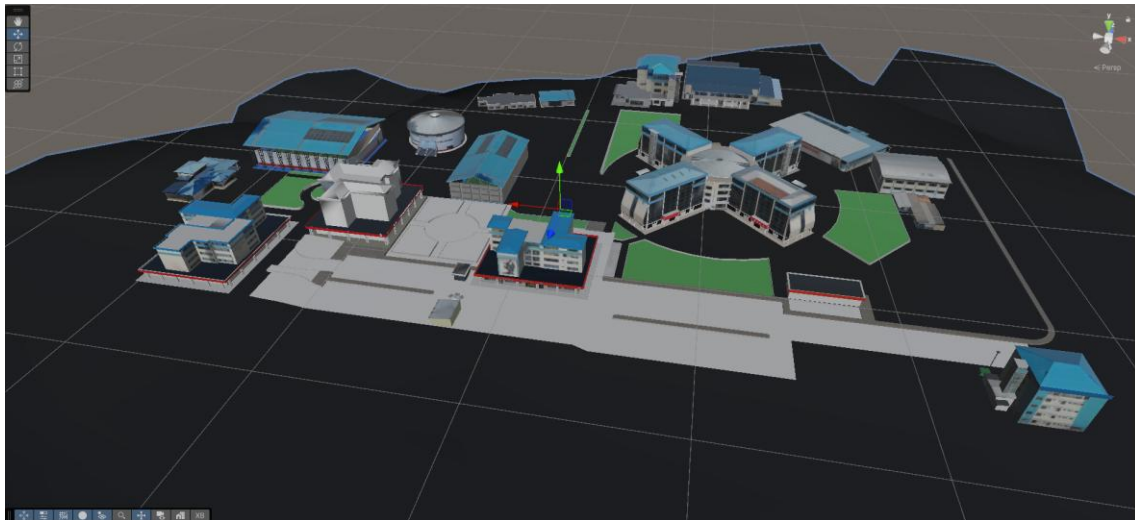


Figura 16 Modelo 3D v5 realizado en colaboración tesistas de Diseño Gráfico

Una vez integrado el modelo en el entorno de desarrollo, se implementaron las interacciones básicas necesarias para su exploración. El usuario puede mover, rotar y acercar o alejar el modelo 3D mediante gestos táctiles, permitiendo una navegación intuitiva y sencilla. Estas interacciones fueron ajustadas para ofrecer una respuesta fluida y consistente durante el uso de la aplicación.

Además, se desarrolló un mecanismo de selección de estructuras que permite al usuario interactuar con los edificios del campus. Al seleccionar una estructura, esta se resalta visualmente, facilitando su identificación dentro del modelo. Asimismo, se habilitó un comportamiento que permite acercar la vista hacia una estructura seleccionada, mejorando su visualización.

Durante este sprint también se realizaron pruebas del modo “sin realidad aumentada (NoAR)”, evaluando diferentes escenarios de visualización con el objetivo de definir una base estable para la presentación del modelo sin el uso de realidad aumentada. Estas pruebas permitieron ajustar la visualización general del modelo y asegurar su correcto funcionamiento en distintos contextos.



Figura 17 Modelo 3D de la UTN en escenario de modo NoAR



Figura 18 Vista de edificios en el modo NoAR

3.2.2 Sprint 2 – Interfaz de usuario y navegación principal

El segundo sprint se enfocó en el diseño, implementación y mejora de la interfaz de usuario de la aplicación, con el objetivo de garantizar una interacción clara, intuitiva

y adaptable a distintos modos de uso. Durante esta etapa, la interfaz pasó por varias iteraciones hasta alcanzar una estructura estable y funcional.

Para el desarrollo de la interfaz se utilizó el sistema de UI de Unity basado en *canvas*, lo cual permitió organizar los elementos visuales de forma flexible. Como resultado de las pruebas realizadas, se definieron tres tipos de interfaz diferenciadas según el contexto de uso de la aplicación:

1. **Interfaz vertical (Portrait)**, utilizada tanto en el modo AR como en el modo NoAR.
2. **Interfaz horizontal (Landscape)**, utilizada igualmente en los modos AR y NoAR.
3. **Interfaz para Street View (StreetViewUI)**, destinada al modo de visualización interior, la cual quedó definida a nivel estructural para su implementación en sprints posteriores.

La aplicación detecta automáticamente la orientación del dispositivo y cambia dinámicamente entre las interfaces vertical y horizontal, asegurando que los elementos se mantengan correctamente distribuidos en pantalla.



Figura 19 Interfaz del modo con Realidad Aumentada (AR) Vertical

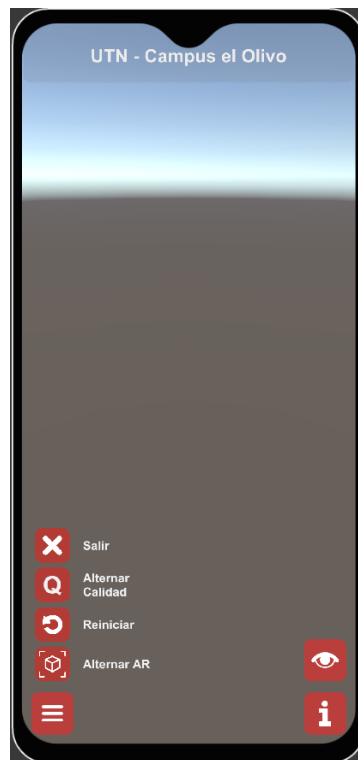


Figura 20 Interfaz del modo sin Realidad Aumentada (NoAR) Vertical

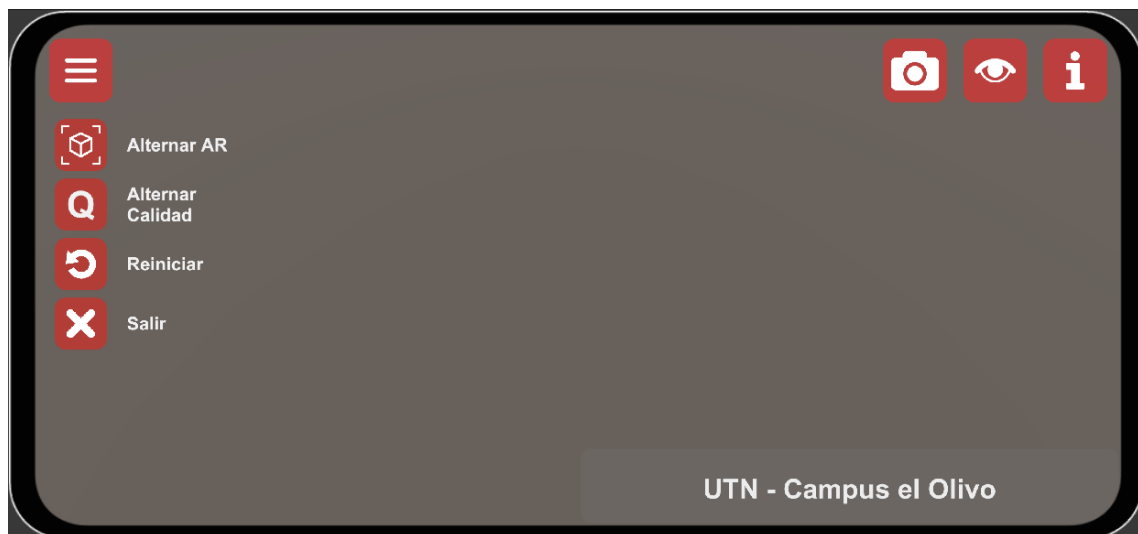


Figura 21 Interfaz del modo con Realidad Aumentada (AR) Horizontal

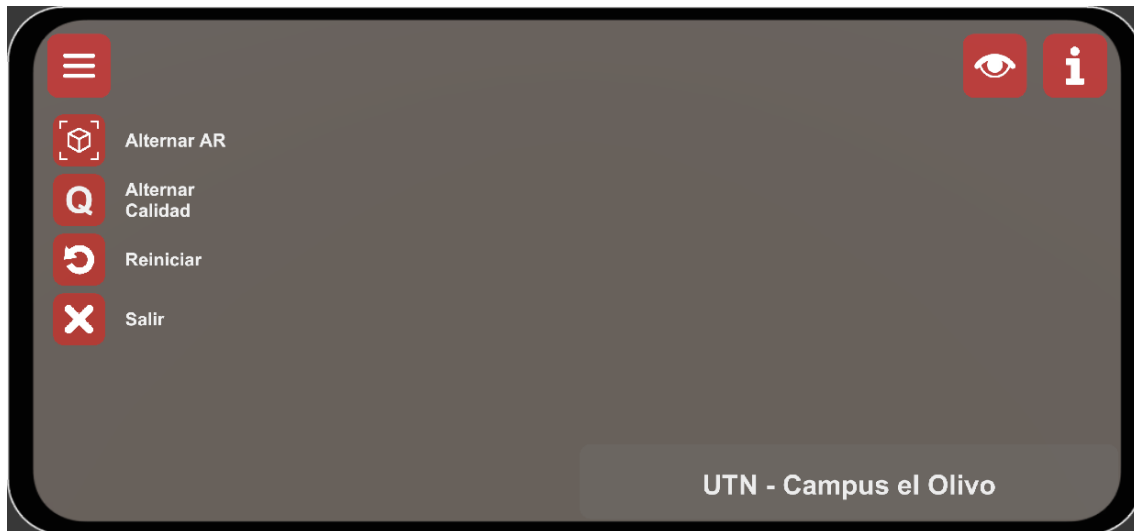


Figura 22 Interfaz del modo sin Realidad Aumentada (NoAR) Horizontal



Figura 23 Interfaz del modo Street View (SV)

En cuanto a la disposición de los controles, en el lado izquierdo de la pantalla se ubicó el menú de opciones principales, desde el cual el usuario puede salir de la aplicación, reiniciar la escena o alternar entre los modos de realidad aumentada y visualización tradicional. En el lado derecho se ubicaron los botones de interacción directa, los cuales permiten visualizar información del edificio seleccionado o ingresar al interior de una estructura.



Figura 24 Opciones del menú en interfaz



Figura 25 Botones de interacción (Captura, Ingreso modo SV, Información)

Adicionalmente, se diseñó e implementó una guía de gestos que se presenta al inicio de la aplicación. Esta guía se muestra mientras el modelo 3D se encuentra en proceso de carga, permitiendo que el usuario comprenda de manera anticipada cómo interactuar con el modelo mediante gestos táctiles, tales como mover, rotar o acercar la vista.



Figura 26 Guía de gestos en pantalla de carga

Finalmente, se implementó la funcionalidad del botón de información. Para utilizar esta opción, el usuario debe seleccionar previamente un edificio dentro del modelo 3D y luego presionar el botón correspondiente. Al hacerlo, se despliega una ventana con información breve del edificio seleccionado, facilitando el acceso a datos relevantes de cada estructura.

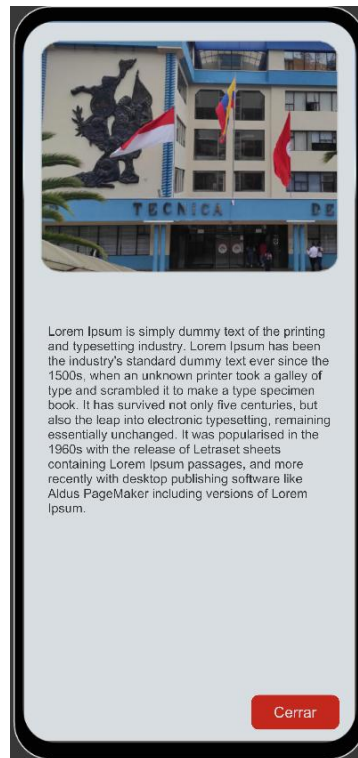


Figura 27 Ventana de información

3.2.3 Sprint 3 – Implementación del modo Street View

El tercer sprint se centró en el desarrollo del modo **Street View (SV)**, el cual permite al usuario recorrer de manera virtual el interior de los edificios de la universidad. Este modo fue diseñado con el objetivo de ofrecer una experiencia de navegación clara, controlada y escalable, similar a la utilizada en plataformas como Google Street View.

Para la implementación de este modo, se desarrolló un sistema basado en **stands y nodos**. Cada *stand* representa una ubicación específica dentro de un edificio y cuenta con una fotografía en 360 grados que permite visualizar el entorno desde ese punto. A su vez, cada stand presenta nodos de navegación que hacen referencia a otros stands, permitiendo al usuario desplazarse entre distintas ubicaciones del interior del edificio.

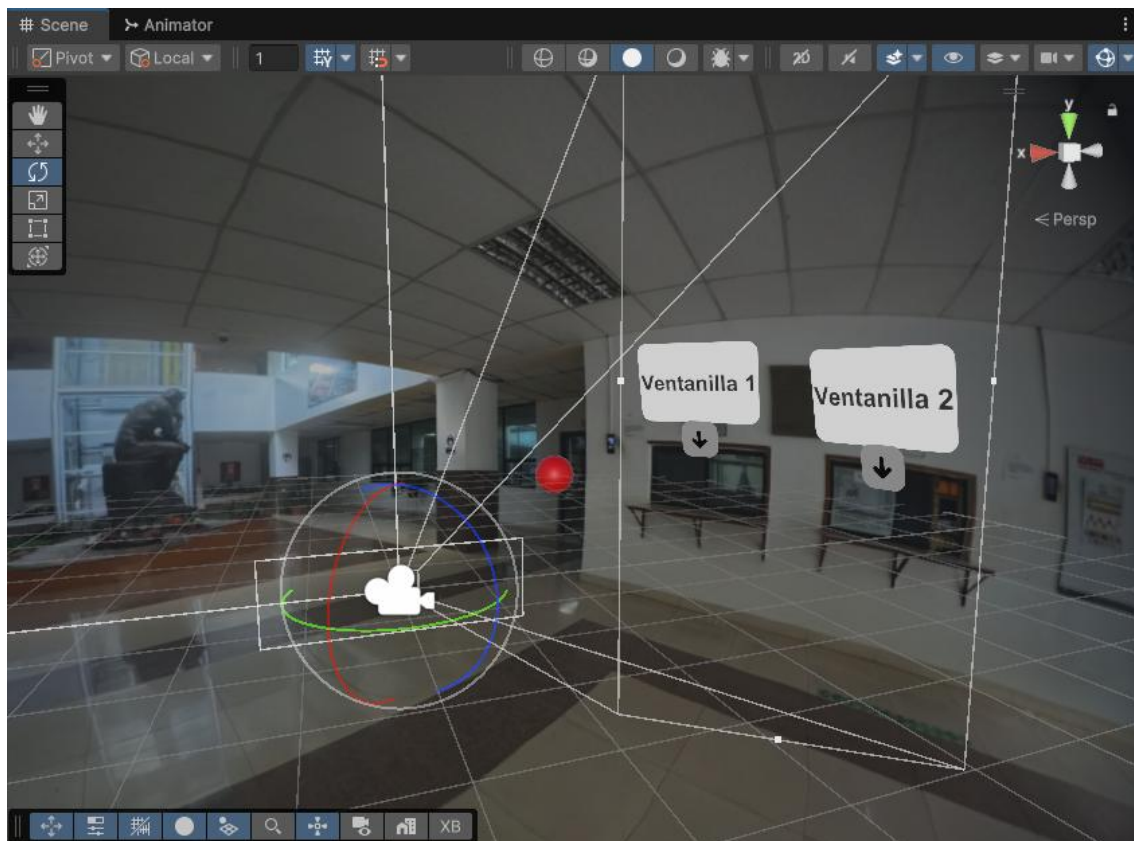


Figura 28 Cámara en modo Street View (SV) en editor de Unity

Se estableció una relación directa entre el edificio seleccionado en el modelo 3D exterior y el modo Street View correspondiente, de manera que, al ingresar al interior, la aplicación cargue automáticamente el recorrido asociado al edificio seleccionado.

En una etapa inicial del desarrollo, todos los nodos de navegación se encontraban activos y la cámara se desplazaba físicamente dentro de la escena de Unity mediante traslaciones. Sin embargo, este enfoque presentó varios inconvenientes, ya que al mantener todos los nodos activos existían casos en los que la cámara podía visualizar nodos no deseados, como los correspondientes a pisos superiores o ubicaciones separadas por paredes. Debido a esto, se rediseñó el sistema para que los nodos se activen y desactiven dinámicamente mediante clics, cambiando la fotografía 360 según la ubicación seleccionada. Este cambio permitió una solución más estable, controlada y fácilmente escalable.

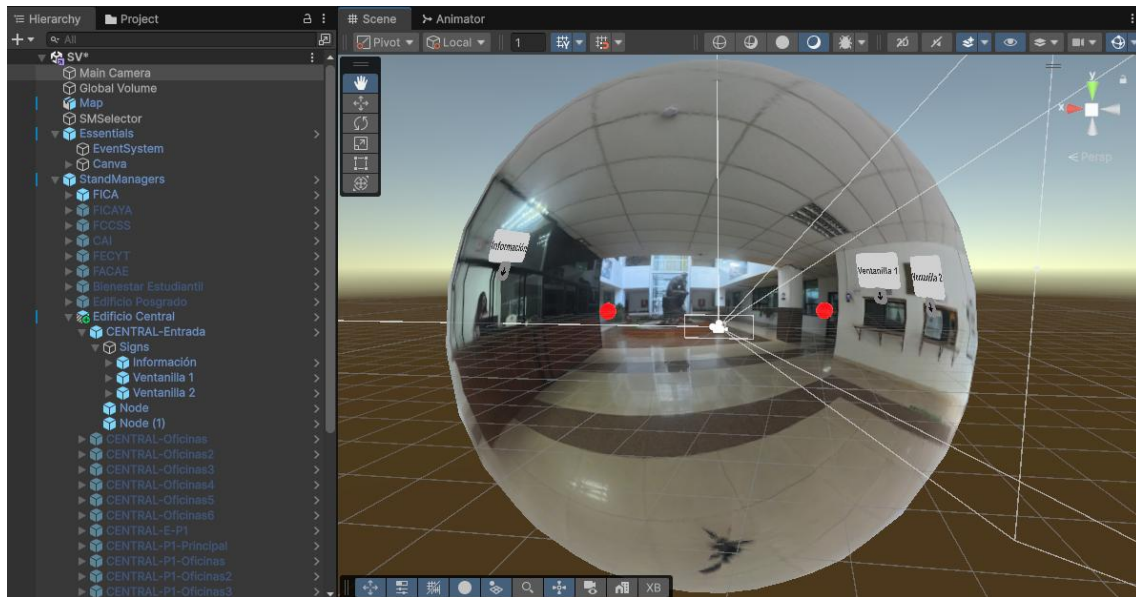


Figura 29 Modo Street View (SV) en editor de Unity

En cuanto a la visualización de las fotografías en 360 grados, inicialmente se utilizó una *skybox* para mostrar las imágenes. No obstante, esta solución presentó limitaciones técnicas y problemas de escalabilidad. Por esta razón, se optó por utilizar una esfera invertida que muestra las fotografías hacia su interior, ubicando la cámara dentro de la misma y cambiando la textura según el stand activo. Este enfoque permitió una visualización más estable y una mejor adaptación del sistema.

Finalmente, se desarrolló un sistema de movimiento de cámara que permite una experiencia similar a la de Google Street View, en la cual el desplazamiento se realiza mediante nodos de navegación en lugar de permitir movimientos libres. Esto garantiza un recorrido ordenado y evita desplazamientos no deseados dentro del entorno virtual.

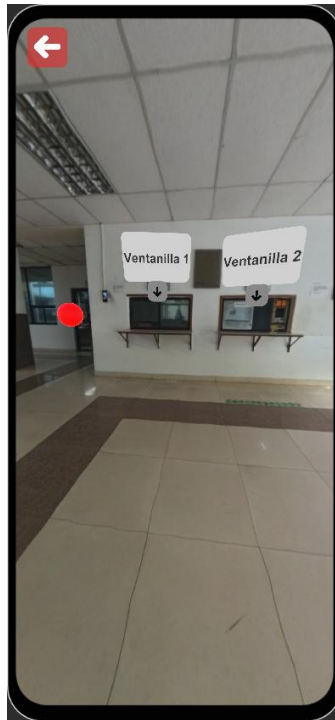


Figura 30 Ejecución del modo Street View (SV)

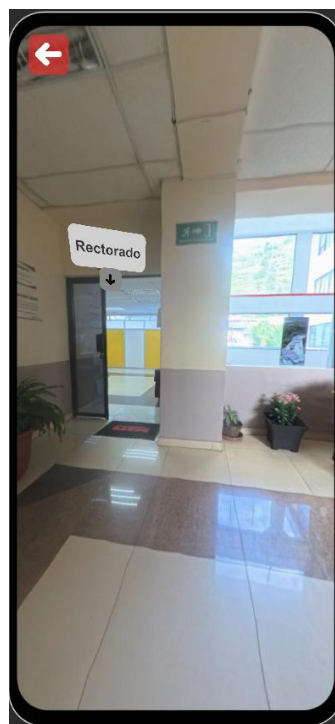


Figura 31 Ejecución del modo Street View (SV)

3.2.4 Sprint 4 – Optimización, seguridad y preparación para despliegue

El cuarto sprint se enfocó en optimizar el contenido para que la aplicación sea viable en dispositivos móviles, mejorar la estabilidad general y preparar la entrega final.

Esta etapa fue importante porque el modelo 3D inicial tenía un tamaño elevado y, además, se debían cumplir restricciones de publicación y seguridad.

En primer lugar, se trabajó en la optimización del modelo 3D del campus, el cual tenía un peso aproximado de **671 MB**. Para reducir este tamaño se realizó una compresión de texturas según la necesidad visual de cada elemento. En general, se disminuyó la resolución de texturas muy grandes, pasando a tamaños más adecuados para móvil (por ejemplo, 1024 píxeles o inferiores en varios casos), logrando un modelo más ligero y práctico para instalar y ejecutar.



Figura 32 Modelo con textura de 4096 pixeles



Figura 33 Modelo con textura de 1024 pixeles

Como el proyecto se desarrolló junto con dos tesistas de Diseño Gráfico, el modelo 3D se mantuvo en constante actualización durante este sprint. Se realizaron cambios y mejoras sucesivas hasta llegar a una versión final más consistente y lista para integrarse en la aplicación.



Figura 34 Modelo 3D v9 (final) realizado en colaboración tesistas de Diseño Gráfico

En paralelo, se corrigieron varios errores y detalles de funcionamiento detectados en la interfaz de usuario. Estos ajustes se aplicaron tanto en el modo NoAR y AR, como en el modo Street View, buscando que los botones, paneles y transiciones funcionen de manera estable y que no existan comportamientos inesperados durante el uso normal.

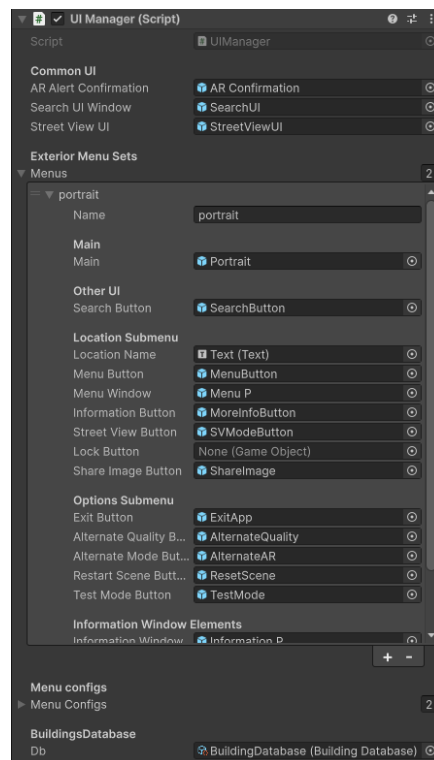


Figura 35 Estructura final del script de gestión de UI en el Inspector de Unity

Un punto importante de este sprint fue la protección del modelo 3D, debido a que el contenido forma parte de un trabajo de titulación y se considera un recurso que no debe quedar accesible de forma directa. Para esto, se implementó un proceso en dos etapas: primero el modelo se empaquetó y cifró antes de integrarlo a la aplicación, y luego, durante la ejecución, la aplicación descifra el contenido para poder cargarlo y usarlo normalmente. Esta parte requirió pruebas adicionales, ya que al cargar el modelo de esta forma podían aparecer problemas visuales que debían corregirse para mantener la apariencia correcta del modelo.

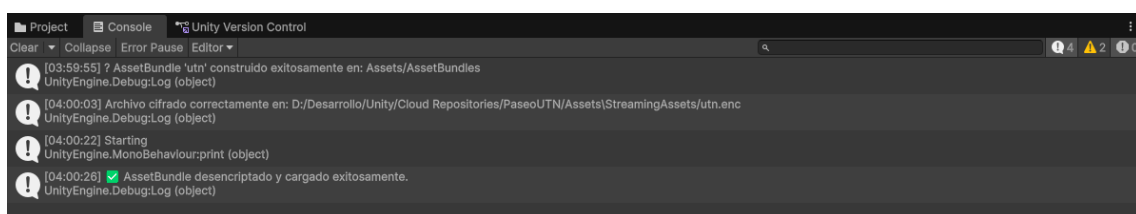


Figura 36 Mensajes en consola del proceso de encriptación y descryptación

Finalmente, considerando el límite de tamaño para publicación en Google Play, se utilizó una estrategia de distribución modular de recursos con Addressables. Esto permitió organizar el contenido de forma que el archivo de publicación no supere el tamaño permitido y que los recursos se gestionen de manera más eficiente según el uso del usuario.

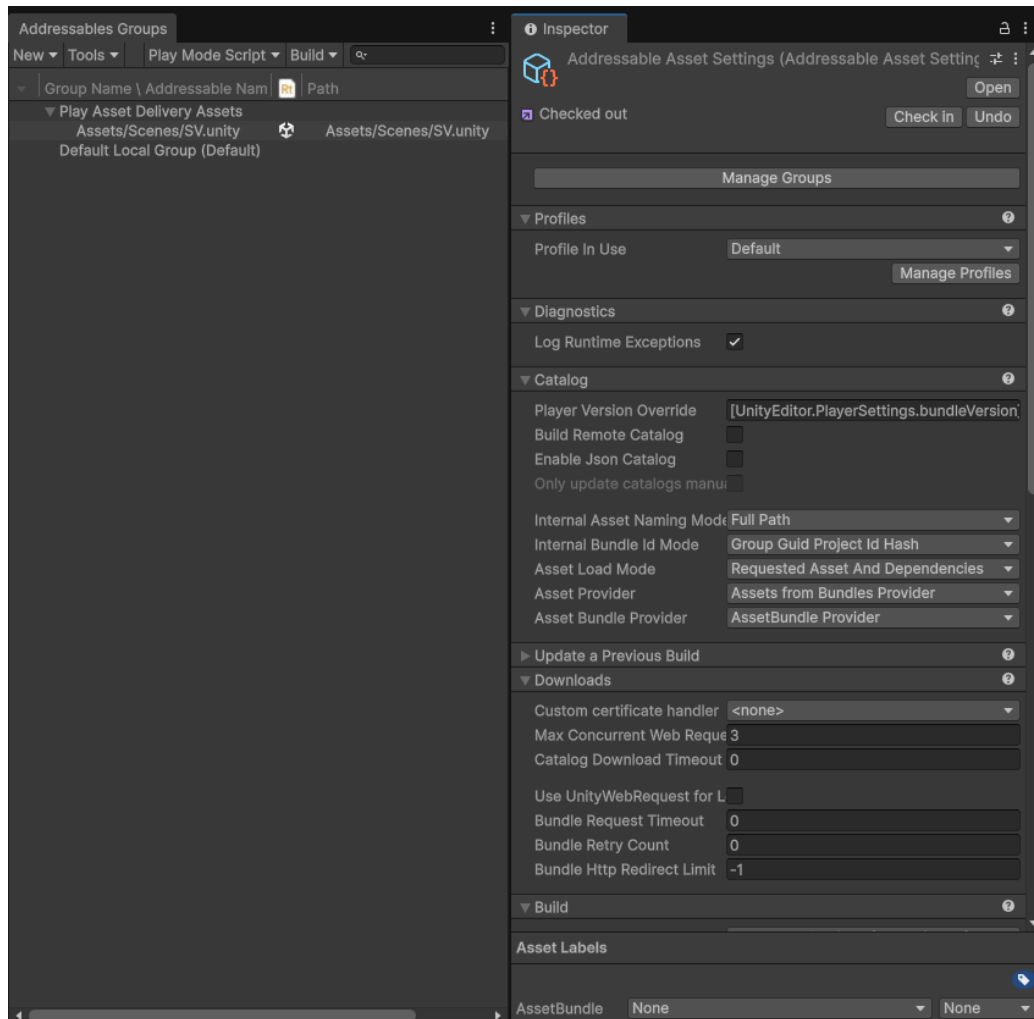


Figura 37 Configuración de Addressables

Como resultado de este sprint, se obtuvo una versión más estable y optimizada de la aplicación, con un modelo 3D más adecuado para móvil, mejoras generales de interfaz y un mecanismo de seguridad que protege el contenido, dejando el sistema listo para su fase final de validación y entrega.

3.3 Fase: Post-juego

La fase pos-juego corresponde a la etapa final del desarrollo del proyecto *Paseo UTN*. En esta fase se realizaron las actividades necesarias para cerrar el proceso de

desarrollo, preparar la versión final de la aplicación, verificar su correcto funcionamiento y realizar la entrega formal del producto.

A diferencia de la fase de juego, en esta etapa no se desarrollaron nuevas funcionalidades, sino que se consolidaron los resultados obtenidos en los sprints anteriores, asegurando la estabilidad, operatividad y disponibilidad del sistema.

3.3.1 Despliegue del producto

Como parte del despliegue del producto, se generó una versión final de la aplicación a partir del proyecto desarrollado en Unity. Durante este proceso se realizaron pruebas finales para verificar que todas las funcionalidades implementadas en los sprints funcionen correctamente en conjunto, tanto en el modo NoAR, AR como en el modo Street View.

Se verificó que la aplicación compile correctamente y que el archivo generado cumpla con los requisitos necesarios para su ejecución en dispositivos móviles. Además, se realizaron pruebas de funcionamiento en dispositivos reales, evaluando aspectos como la carga del modelo 3D, la interacción con la interfaz, la navegación entre modos y la estabilidad general del sistema.

Se puso especial atención al tamaño final de la aplicación, asegurando que el archivo de publicación no supere el límite permitido por Google Play. Gracias al uso de técnicas de optimización y a la carga modular de recursos, la aplicación logró cumplir con este requerimiento sin afectar la experiencia del usuario.

3.3.2 Verificación de funcionamiento

Previo a la entrega final, se realizó una verificación general del funcionamiento de la aplicación. Esta verificación consistió en comprobar que las principales funcionalidades definidas en las historias de usuario se encuentren operativas y no presenten errores críticos.

Entre los aspectos verificados se incluyen la correcta visualización del modelo 3D, las interacciones táctiles, el cambio entre modos de visualización, el acceso a la información de los edificios, el recorrido interior mediante Street View y la estabilidad de la interfaz de usuario. Asimismo, se comprobó que el sistema de carga segura del modelo funcione correctamente durante la ejecución de la aplicación.

Estas pruebas permitieron confirmar que la aplicación se encuentra en un estado funcional y estable, apto para su entrega y posterior evaluación.

3.3.3 Entrega del producto

Una vez finalizado el proceso de despliegue y verificación, se procedió a la entrega del producto. La entrega consistió en la presentación de la versión final y funcional de la aplicación *Paseo UTN*, desarrollada conforme a los objetivos planteados en el proyecto.

El producto entregado incluye la aplicación móvil operativa, así como el proyecto desarrollado en el entorno de Unity, quedando disponible para la institución como una herramienta de apoyo para la visualización y exploración virtual del campus universitario.

Con esta entrega se da por concluida la fase de desarrollo del proyecto, quedando el sistema listo para su posterior evaluación mediante criterios de calidad en uso y pruebas de usabilidad.

3.3.4 Cierre de la fase pos-juego

La fase pos-juego permitió consolidar el trabajo realizado durante el desarrollo del proyecto, asegurando que la aplicación cumpla con los requerimientos definidos y se encuentre lista para su evaluación y validación. Esta etapa marca el cierre del proceso de desarrollo bajo la metodología SCRUM, dando paso a la fase de resultados y análisis del sistema.

CAPÍTULO IV

RESULTADOS Y ANÁLISIS

4.1 Modelo de evaluación aplicado (ISO/IEC 25022 – Calidad en uso)

Para evaluar la aplicación *Paseo UTN*, se utilizó un modelo basado en el estándar ISO/IEC 25022, el cual permite medir cómo se comporta un sistema cuando lo usan personas reales. En este caso, el objetivo fue revisar si la aplicación es fácil de usar, si las tareas se pueden completar sin problemas y qué tan buena fue la experiencia general de los participantes.

El modelo se divide en tres partes principales:

- Eficacia: si los usuarios logran completar las tareas propuestas.

- Eficiencia: cuánto tiempo se demoran en completar esas tareas.
- Satisfacción: qué tan cómoda y agradable les pareció la aplicación.

Para obtener estos datos se aplicaron dos instrumentos:

1. Taller práctico, donde cada participante realizó tareas definidas dentro de la aplicación y se registró si las completó, si tuvo errores y cuánto tiempo le tomó.
2. Cuestionario, para conocer la percepción del usuario después de usar la aplicación.

En la Tabla 22 se muestra un resumen de las métricas utilizadas, cómo se midieron y de dónde se obtuvo cada valor.

Tabla 22 Modelo de evaluación basado en ISO/IEC 25022

<i>Característica</i>	<i>Métrica</i>	<i>Qué se midió</i>	<i>Fórmula usada</i>
Eficacia	Finalización de tareas	Cuántas tareas se completaron correctamente	$(\text{Tareas finalizadas} / \text{Tareas totales}) * 100$
	Tareas sin errores	Cuántas tareas se realizaron sin fallos	$(\text{Tareas sin error} / \text{Tareas totales}) * 100$
Eficiencia	Tiempo promedio por tarea	Tiempo promedio que tomó completar cada tarea	$(\text{Suma de tiempos de tareas} / \text{Número de tareas})$
Satisfacción	Puntaje promedio del cuestionario	Promedio de la calificación de los usuarios (1 a 5)	$(\text{Suma de puntajes} / \text{Número de respuestas})$
	Corrección de preguntas negativas	Ajuste para preguntas donde “más alto” significa algo negativo	$6 - x$

El cuestionario aplicado incluye preguntas redactadas en dos sentidos:

Preguntas positivas, donde un valor alto (4 o 5) significa una mejor percepción.
Por ejemplo: “*la aplicación es fácil de utilizar*”.

Preguntas negativas, donde un valor alto (4 o 5) significa una peor percepción.
Por ejemplo: “*la aplicación es compleja*” o “*la aplicación resulta confusa*”.

Para que el análisis sea correcto y todas las preguntas se interpreten en la misma dirección, se aplicó el mismo método común de corrección por polaridad, donde las preguntas negativas se invierten antes de calcular el resultado general. De esta forma, todas las respuestas quedan con el mismo significado:

5 = mejor percepción

1 = peor percepción

La corrección se aplicó con la siguiente fórmula:

$$\text{Puntaje corregido} = 6 - x$$

Donde X es el valor original en escala de 1 a 5.

4.2 Población y muestra

Para realizar la evaluación de la aplicación Paseo UTN, se trabajó con una muestra de 46 participantes ($n = 46$). Los encuestados fueron estudiantes universitarios de los últimos niveles de la carrera de Ingeniería en Software, ya que este grupo cuenta con mayor experiencia en el uso de herramientas tecnológicas y facilita la comprensión de las actividades planteadas durante la prueba.

La muestra utilizada corresponde a la misma que fue aplicada en la evaluación de los módulos del aplicativo UTN Móvil, debido a que ambas pruebas se desarrollaron dentro del mismo proceso experimental, manteniendo condiciones similares y permitiendo realizar una evaluación uniforme.

Cada participante realizó un taller práctico compuesto por 8 tareas, donde se registró información como el tiempo de ejecución, presencia de errores y finalización de cada actividad. Posteriormente, se aplicó un cuestionario de percepción con escala de 1 a 5, con el objetivo de conocer la opinión del usuario en aspectos como facilidad de uso, complejidad, aprendizaje y seguridad durante la utilización de la aplicación.

4.3 Resultados del Taller práctico

El taller práctico se aplicó con el objetivo de obtener resultados objetivos sobre el uso del módulo Paseo UTN, considerando el desempeño real de los usuarios durante la ejecución de tareas dentro de la aplicación. Para ello, se definieron ocho tareas que representan acciones básicas e importantes del sistema, como interacción con el modelo 3D, visualización de información, navegación por el submenú y cambio de modo.

Durante la prueba se registraron tres aspectos principales:

1. Finalización de tareas,
2. Errores durante la ejecución,
3. Tiempo empleado por tarea.

Estos datos permiten analizar los componentes de eficacia y eficiencia definidos en el modelo de calidad en uso basado en ISO/IEC 25022.

4.3.1 Eficacia

La eficacia se evaluó mediante el porcentaje de tareas finalizadas por los participantes. Los resultados muestran que, en general, la aplicación permitió completar

las tareas con un alto nivel de éxito, ya que la mayoría alcanzó una tasa de finalización cercana al 97.7%.

Sin embargo, se identificaron dos tareas con la menor tasa de finalización:

- Tarea 3: Seleccionar el Edificio Central y abrir la ventana de información (93.2%).
- Tarea 4: Seleccionar el Edificio de Posgrado y abrir la ventana de información (93.2%).

En comparación con el resto de las tareas, estos valores reflejan que las actividades relacionadas con la selección de edificios y acceso a la información fueron las que presentaron mayor probabilidad de no completarse, lo cual puede estar vinculado a la facilidad de identificación del elemento seleccionado o al acceso a la ventana de información.

De manera general, la alta tasa de finalización evidencia que el módulo evaluado permite cumplir los objetivos planteados en el taller y que el flujo de interacción principal es comprensible para la mayoría de los usuarios.

4.3.2 Errores durante la ejecución

El análisis de errores permite identificar las tareas donde los participantes presentaron mayores dificultades o inconvenientes durante la ejecución. En términos generales, los resultados evidencian una baja frecuencia de errores en la mayoría de las tareas, lo cual sugiere que la aplicación mantiene un comportamiento estable durante el uso normal.

La tarea que presentó la mayor frecuencia de error fue:

- Tarea 6: Alternar AR (11.4%).

Este valor indica que el cambio de modo hacia realidad aumentada fue la acción que mayor cantidad de fallos presentó durante el taller. Esto puede estar relacionado con que dicha opción se encuentra dentro del submenú de la aplicación y requiere que el usuario ubique correctamente el botón antes de ejecutar la acción.

Otras tareas que presentaron errores, aunque en menor proporción, fueron:

- Tarea 4: (6.8%)
- Tarea 1, 3 y 7: (4.5% cada una)

Por otro lado, se registraron tareas sin errores durante la ejecución:

- Tarea 2, 5 y 8: (0%)

En conjunto, estos resultados reflejan que los errores se presentaron principalmente en tareas relacionadas con la navegación de opciones específicas, mientras que las tareas más directas o intuitivas tuvieron menor probabilidad de fallos.

4.3.3 Eficiencia temporal

La eficiencia se evaluó mediante el tiempo promedio requerido para completar cada tarea del taller. Esta medición permite identificar cuáles actividades demandan mayor esfuerzo o tiempo por parte del usuario.

Las tareas con mayor tiempo promedio fueron:

- Tarea 1: Rotar, trasladar y escalar modelo 3D (50.48 s)
- Tarea 2: Centrar edificio usando gestos (40.14 s)
- Tarea 4: Abrir información del edificio de Posgrado (33.23 s)

Estos tiempos son coherentes, especialmente en el caso de las tareas iniciales, ya que implican familiarización con la interacción en 3D y el uso de gestos de navegación dentro del modelo.

En contraste, las tareas con menor tiempo promedio fueron:

- Tarea 5: Reiniciar ubicación (13.57 s)
- Tarea 8: Salir de la aplicación (17.89 s)
- Tarea 7: Alternar No AR (21.80 s)

Finalmente, considerando la suma de los tiempos promedio de las ocho tareas, el tiempo total aproximado requerido para completar el taller práctico fue de:

234.66 segundos (~3.9 minutos)

Este resultado permite concluir que el conjunto de actividades puede ejecutarse en un tiempo reducido, lo cual es positivo para un módulo que incluye interacción con un modelo 3D, navegación entre elementos y uso de funciones específicas dentro de la interfaz.

4.4 Resultados del cuestionario

Luego de finalizar el taller práctico, se aplicó un cuestionario con escala tipo Likert (1–5) para conocer la percepción de los usuarios sobre el módulo Paseo UTN,

considerando criterios relacionados con facilidad de uso, complejidad, aprendizaje y seguridad.

Es importante mencionar que el cuestionario incluyó preguntas formuladas de manera positiva y negativa. Por esta razón, para obtener resultados comparables, las preguntas negativas fueron corregidas invirtiendo su valor (mediante la expresión: Puntaje invertido = $6 - x$)

4.4.1 Resultados por pregunta (Likert 1–5)

Los resultados corregidos muestran una percepción mayormente positiva hacia el módulo evaluado. A continuación, se describen los valores más relevantes por ítem:

- Frecuencia de uso (P1): promedio corregido 3.37
Indica que los usuarios consideran que podrían utilizar la aplicación con cierta frecuencia.
- Complejidad (P2): promedio corregido 3.58
Sugiere que la aplicación no fue considerada compleja para la mayoría.
- Facilidad de uso (P3): promedio corregido 3.68
Refleja que el sistema es percibido como relativamente fácil de utilizar.
- Necesidad de apoyo experto (P4): promedio corregido 4.44
Muestra que, en general, los usuarios no consideran necesario depender de un experto para usar la aplicación.
- Inconsistencias (P5): promedio corregido 4.38
Indica que los participantes percibieron baja presencia de inconsistencias en el uso.
- Facilidad de aprendizaje (P6): promedio corregido 4.00
Señala que la aplicación es fácil de aprender para la mayoría.
- Confusión al usar (P7): promedio corregido 4.07
Refleja que los usuarios no la consideran confusa en general.
- Sistema confuso (P8): promedio corregido 4.53
Este fue uno de los valores más altos, reforzando la percepción de claridad del sistema.

- Seguridad al utilizar (P9): promedio corregido 4.40
Indica que la mayoría se sintió segura usando el módulo evaluado.
- Requerimientos técnicos previos (P10): promedio corregido 4.00
Sugiere que los usuarios no consideran necesarios conocimientos técnicos avanzados para usarla correctamente.

En general, se observa que los valores más altos se concentran en los ítems relacionados con confianza, consistencia y claridad del sistema, lo cual es positivo para la experiencia de uso.

4.4.2 Resultado por subcaracterística

Para una mejor interpretación, los ítems del cuestionario se agruparon según el tipo de aspecto evaluado, manteniendo la lógica aplicada en la tesis de referencia.

- Facilidad de uso y aprendizaje: P3, P6
Los resultados indican una valoración positiva, ya que los promedios se mantienen cercanos o superiores a 4 en algunos casos.
- Complejidad y confusión percibida: P2, P7, P8
Los valores corregidos sugieren que los usuarios percibieron la aplicación como poco compleja y con baja confusión durante el uso.
- Autonomía del usuario (sin apoyo experto): P4, P10
Se evidencia que los participantes consideran que el módulo puede ser utilizado sin necesidad de apoyo especializado, lo cual es favorable.
- Consistencia del sistema: P5
La percepción indica que el sistema presenta pocas inconsistencias, lo cual puede asociarse a un funcionamiento estable.
- Seguridad en el uso: P9
Se obtuvo un valor alto, mostrando que el usuario se siente confiado al interactuar con la aplicación.

4.5 Validación del instrumento y confiabilidad

Con el fin de asegurar que los resultados del cuestionario sean consistentes y confiables, se realizó la validación estadística del instrumento utilizado, siguiendo el enfoque aplicado en la tesis base.

4.5.1 Prueba de normalidad

Se aplicó una prueba de normalidad con el objetivo de conocer la distribución de los datos recolectados. Esta validación permite determinar si es adecuado aplicar análisis paramétricos o si es más conveniente utilizar análisis descriptivo y no paramétrico.

En este trabajo, considerando el tipo de escala Likert y el objetivo de evaluación, los resultados se interpretaron principalmente mediante estadística descriptiva (promedios y porcentajes).

4.5.2 Confiabilidad del cuestionario (Alfa de Cronbach)

Para medir la confiabilidad del instrumento se utilizó el coeficiente Alfa de Cronbach, el cual permite analizar si las preguntas del cuestionario tienen coherencia interna.

La interpretación usual del valor del Alfa de Cronbach es la siguiente:

- ≥ 0.70 : aceptable
- ≥ 0.80 : buena
- ≥ 0.90 : excelente

El cuestionario aplicado corresponde al mismo utilizado en la evaluación de otros módulos del aplicativo UTN Móvil, por lo cual se mantiene coherencia metodológica en la aplicación y análisis del instrumento.

4.6 Resultado global de Calidad en Uso

Para obtener un resultado general de calidad en uso del módulo Paseo UTN, se integraron los valores de Eficacia, Eficiencia y Satisfacción, de acuerdo con el modelo basado en ISO/IEC 25022. Para este cálculo se aplicó una ponderación que permite dar mayor importancia a los resultados observables del taller práctico, ya que estos reflejan directamente el desempeño del usuario al ejecutar tareas reales dentro de la aplicación.

Los pesos definidos fueron:

- Eficacia: 0.40
- Eficiencia: 0.30
- Satisfacción: 0.30

La fórmula aplicada fue:

$$\text{Calidad en uso} = (0.40 \cdot \text{Eficacia}) + (0.30 \cdot \text{Eficiencia}) + (0.30 \cdot \text{Satisfacción})$$

En este caso, la Eficacia se calculó a partir del promedio de finalización de las tareas del taller práctico, obteniendo un valor aproximado de 96.6%. La Satisfacción se obtuvo a partir del promedio corregido del cuestionario (4.012/5), equivalente a 80.2%. Finalmente, la Eficiencia temporal se estimó comparando el tiempo total promedio del taller (234.66 s) con un tiempo esperado de referencia (300 s), resultando en 78.2%.

Como resultado final, se obtuvo un valor global de calidad en uso de: **86.2%**, lo cual representa un desempeño general favorable del módulo evaluado, considerando tanto el cumplimiento de tareas como la percepción del usuario.

4.7 Análisis e interpretación general

En base a los resultados obtenidos, se puede concluir que el módulo Paseo UTN presenta un desempeño positivo en cuanto a la ejecución de tareas, ya que la mayoría de los usuarios logró finalizar las actividades propuestas con un alto porcentaje de éxito y con una baja presencia de errores.

Las principales dificultades se observaron en tareas relacionadas con el acceso al submenú y el cambio de modo (Alternar AR), lo cual sugiere que se trata de una función que podría mejorarse mediante indicaciones más claras, una ubicación más visible o una interacción más directa.

Por otro lado, los resultados subjetivos del cuestionario reflejan que los usuarios percibieron la aplicación como fácil de usar, segura y con baja complejidad, lo cual respalda los resultados del taller y muestra coherencia entre la medición objetiva y la percepción del usuario.

Conclusiones y recomendaciones

Conclusiones

- Se desarrolló e implementó la aplicación móvil Paseo UTN, permitiendo la exploración del campus mediante un modelo 3D, la visualización de información de edificios y el acceso a opciones de interacción desde la interfaz.
- La evaluación objetiva mediante el taller práctico evidenció un alto nivel de eficacia, ya que el promedio de finalización de las tareas fue aproximadamente 96.6%, lo cual demuestra que la mayoría de usuarios logró completar las acciones planteadas.
- Los errores durante el taller se mantuvieron en un nivel bajo en la mayoría de tareas, sin embargo, se identificó que la tarea relacionada con alternar al modo AR presentó la mayor frecuencia de error (11.4%), lo que indica que esa función requiere mejoras en claridad o acceso desde la interfaz.
- En términos de eficiencia temporal, se registró un tiempo promedio total de 234.66 segundos para completar las ocho tareas, lo cual refleja que el módulo puede ser utilizado en un tiempo reducido y sin necesidad de procesos extensos para el usuario.
- Los resultados subjetivos obtenidos mediante el cuestionario indicaron una percepción positiva del módulo, alcanzando un promedio corregido global de 4.012/5, destacando especialmente la facilidad de aprendizaje, la baja confusión durante el uso y la seguridad percibida al utilizar la aplicación.
- Al integrar eficacia, eficiencia y satisfacción mediante ponderación, se obtuvo un resultado global de calidad en uso aproximado de 86.2%, lo cual evidencia que la aplicación cumple de forma satisfactoria con los objetivos planteados en el contexto de orientación y exploración del campus.

Recomendaciones

- Se recomienda mejorar la opción “Alternar AR”, ya que fue la tarea con mayor porcentaje de errores, incorporando una ubicación más visible en la interfaz o una señalización más clara para el usuario.
- Se recomienda incluir apoyos visuales breves dentro del módulo, como una guía inicial o indicaciones rápidas, especialmente para la interacción con el modelo 3D, debido a que esta sección presentó los tiempos más altos en comparación con el resto.
- Se recomienda ajustar la visualización o selección de edificios al abrir ventanas de información, debido a que las tareas asociadas a esta acción fueron las que presentaron menor porcentaje de finalización, lo cual puede mejorar la experiencia del usuario en la exploración del campus.
- Se recomienda ampliar futuras evaluaciones con una muestra más diversa, incluyendo usuarios externos a la carrera o visitantes del campus, con el fin de validar el comportamiento del sistema con perfiles distintos al evaluado en este estudio.
- Se recomienda mantener el modelo de evaluación basado en ISO/IEC 25022 para futuras versiones del proyecto, ya que permite medir de forma ordenada el

desempeño del sistema y facilita comparar resultados cuando se agreguen nuevas funciones.

- Se recomienda continuar con la optimización general del sistema y del modelo 3D para asegurar una mejor fluidez en diferentes dispositivos móviles, especialmente en aquellos con menor capacidad de procesamiento.

Referencias Bibliográficas

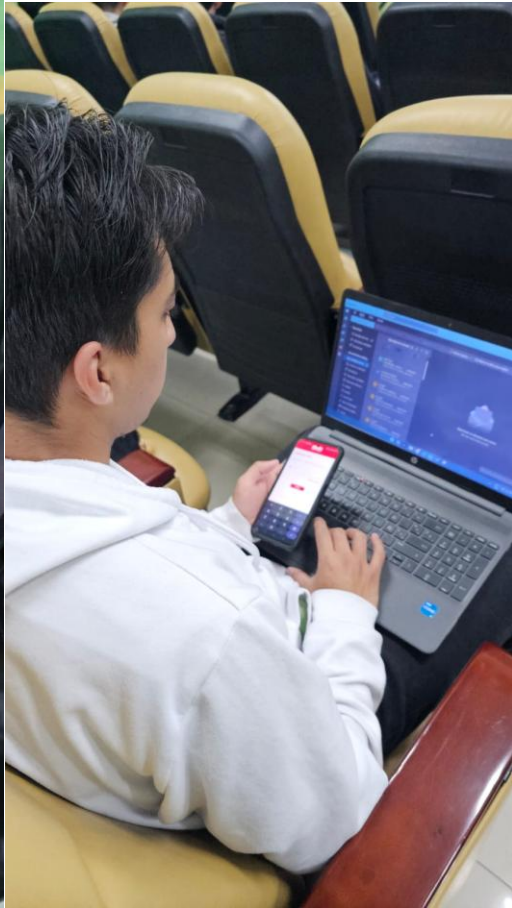
- [1] «AGENDA DE TRANSFORMACIÓN DIGITAL ECUADOR – Ministerio de Telecomunicaciones y de la Sociedad de la Información», Ministerio de Telecomunicaciones y de la Sociedad de la Información. Accedido: 30 de octubre de 2024. [En línea]. Disponible en: <https://www.telecomunicaciones.gob.ec/agenda-de-transformacion-digital-ecuador/>
- [2] «Proyecto Typical-Food», Arutn. Accedido: 30 de octubre de 2024. [En línea]. Disponible en: <https://realidadutn.wixsite.com/arutn/typical-food>
- [3] «IbarraAdventure», Arutn. Accedido: 30 de octubre de 2024. [En línea]. Disponible en: <https://realidadutn.wixsite.com/arutn/ibarraadventure>
- [4] D. Lopez, «Diseño de un recorrido virtual del campus universitario como propuesta para implementación en el portal web de la universidad nacional de Loja», Universidad Nacional de Loja, 2017. [En línea]. Disponible en: <https://dspace.unl.edu.ec/jspui/bitstream/123456789/19216/1/L%C3%B3pez%20Lalangui%2C%20Diego%20Patricio.pdf>
- [5] «Tour Virtual 360 Sede Cuenca - UPS». Accedido: 30 de octubre de 2024. [En línea]. Disponible en: <https://ww1.ups.edu.ec/web/guest/tour-virtual-360-sede-cuenca1>
- [6] «Virtual Tour | Harvard». Accedido: 30 de octubre de 2024. [En línea]. Disponible en: <https://college.harvard.edu/admissions/explore-harvard/virtual-tour>
- [7] «Virtual Tour University of Oklahoma». Accedido: 30 de octubre de 2024. [En línea]. Disponible en: <http://www.ou.edu/admissions/visit/virtual-tour.html>
- [8] C. R. Bello, «La realidad aumentada: lo que debemos conocer», *Tecnol. Investig. Acad.*, vol. 5, n.º 2, Art. n.º 2, nov. 2017.
- [9] F. Telefónica, *Realidad Aumentada: una nueva lente para ver el mundo*. Fundación Telefónica, 2011.
- [10] I. M. M. Bohórquez, «Realidad aumentada y aplicaciones», *Tecnol. Investig. Acad.*, vol. 6, n.º 1, Art. n.º 1, mar. 2018.
- [11] P. J. Sánchez Bermejo, «Modelado 3D correcto.», 2020, [En línea]. Disponible en: <https://hdl.handle.net/20.500.14352/10077>
- [12] R. Rosales, D. Murillo, y R. Miguelena, «Modelado y Animación 3D», *El Tecnológico*, vol. 27, n.º 1, Art. n.º 1, abr. 2017.
- [13] «Nivel de detalle (LOD)», *Wikipedia, la enciclopedia libre*. 24 de septiembre de 2023. Accedido: 5 de enero de 2025. [En línea]. Disponible en: [https://es.wikipedia.org/w/index.php?title=Nivel_de_detalle_\(LOD\)&oldid=153989006](https://es.wikipedia.org/w/index.php?title=Nivel_de_detalle_(LOD)&oldid=153989006)
- [14] «Que es un Tour Virtual?», 3DVista. Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://www.3dvista.com/es/kb/faqs-es/tours-virtuales/que-es-un-tour-virtual/>
- [15] C. Haba, «¿Qué es un Tour Virtual? La nueva herramienta de Marketing 4.0», Innoarea Projects. Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://innoarea.com/noticias/que-es-tour-virtual/>
- [16] M. L. Mañas, *Unity 3D*. Marcombo, 2019. [En línea]. Disponible en: <https://books.google.es/books?hl=es&lr=&id=3EtOEAAAQBAJ&oi=fnd&pg=PT5&dq=Unity+3D&ots=uWl6d6XA7N&sig=9Ych2Jc2VJhQnIR3D3ixiEuTGzY#v=onepage&q=Unity%203D&f=false>

- [17] U. Technologies, «Unity - Manual: Introduction to scenes». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://docs.unity3d.com/6000.0/Documentation/Manual/CreatingScenes.html>
- [18] U. Technologies, «Unity - Manual: Asset workflow». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://docs.unity3d.com/6000.0/Documentation/Manual/AssetWorkflow.html>
- [19] U. Technologies, «Unity - Manual: AR development in Unity». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://docs.unity3d.com/6000.0/Documentation/Manual/AROverview.html>
- [20] «About Blender - Blender 4.3 Manual». Accedido: 6 de enero de 2025. [En línea]. Disponible en: https://docs.blender.org/manual/en/latest/getting_started/about/index.html
- [21] «Uso de Blender para modelado de mobiliario». Accedido: 17 de enero de 2025. [En línea]. Disponible en: <https://digitalmodels3d.com/uso-de-blender-para-modelado-de-mobiliario/>
- [22] K. S. & J. Sutherland, «La Guía Definitiva de Scrum: Las Reglas del Juego», nov. 2020, Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://repositorio.uvm.edu.ve/handle/123456789/59>
- [23] N. Acerenza *et al.*, «Una metodología para desarrollo de videojuegos», en *38o JAIIO-Simposio Argentino de Ingeniería de Software*. https://www.fing.edu.uy/sites/default/files/biblio/22811/asse_2009_16.pdf, 2009.
- [24] «NORMAS ISO 25000». Accedido: 17 de enero de 2025. [En línea]. Disponible en: <https://iso25000.com/index.php/normas-iso-25000>
- [25] «ISO/IEC 25022:2016», ISO. Accedido: 17 de enero de 2025. [En línea]. Disponible en: <https://www.iso.org/standard/35746.html>
- [26] A. W. Liang, N. Wahid, y T. Gusman, «Virtual Campus Tour Application through Markerless Augmented Reality Approach», *JOIV Int. J. Inform. Vis.*, vol. 5, n.º 4, pp. 354-359, dic. 2021, doi: 10.30630/joiv.5.4.743.
- [27] F. J. Pamungkas, P. Hendradi, A. Setiawan, y S. N. M. Mohamad, «VIRTUAL CAMPUS TOUR BERBASIS WEB MENGGUNAKAN TEKNOLOGI AUGMENTED REALITY», *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 13, n.º 3, Art. n.º 3, jul. 2024, doi: 10.30591/smartcomp.v13i3.6792.
- [28] R. R. Rembulan, «Aplikasi Virtual Tour Islamic Center Berbasis Android», *J. Inform. Dan Rekayasa Perangkat Lunak*, vol. 1, n.º 2, pp. 203-214, dic. 2020.
- [29] J. P. Gois *et al.*, «Memórias Tridimensionais do Grande ABC: Tour Virtual 3D pela Vila de Paranapiacaba», 2016, 2016.
- [30] «Virtual Tour With Voice Assistant using Extended Reality», *ResearchGate*, dic. 2024, doi: 10.35940/ijeat.E4127.0612523.
- [31] A. Handore, Y. Suryawanshi, A. Shaikh, U. Ahire, y C. Patil, «Virtual Reality Based Virtual Tour of College Using Unity 3D», *IJRESM*, vol. 3, n.º 6.
- [32] I. Cibilić, V. Poslončec-Petrić, y K. Tominić, «Implementing Augmented reality in Tourism», *Proc. ICA*, vol. 4, pp. 1-5, dic. 2021, doi: 10.5194/ica-proc-4-21-2021.
- [33] T. Montenegro, «Exploracion en Arquitectura: Un proceso basado en Blender», UNIANDES, 2007. [En línea]. Disponible en: <https://repositorio.uniandes.edu.co/server/api/core/bitstreams/a7b5177c-2393-484f-a9c3-2d70edcf15ff/content>
- [34] «Modelo original de Scrum para desarrollo de software», Scrum Manager BoK. Accedido: 12 de enero de 2026. [En línea]. Disponible en: https://www.scrummanager.com/bok/index.php?title=Modelo_original_de_Scrum_para_desarrollo_de_software
- [35] M. M. U. Nikual, «La elicitación de requisitos en el contexto de un proyecto software», *Ing. USBMed*, vol. 2, n.º 2, pp. 25-29, dic. 2011, doi: 10.21500/20275846.255.

Anexos

Anexo A. Fotografías de la aplicación del taller práctico y encuesta SUS





Anexo B. Presentación de proyectos y equipo UTN Móvil a la máxima autoridad UTN.



Anexo C. Reuniones equipo UTN Móvil







Anexo D. Presentación de UTN Móvil a un curso



Anexo E. Taller práctico aplicado en la evaluación del aplicativo Paseo UTN

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE SOFTWARE

UTN Móvil



INDICACIONES:

1

Complete la tabla con la hora de inicio y la hora de fin de cada tarea

3

Indique si la aplicación mostró algún error durante la tarea (Si/No)

4

Indique si completó la tarea con éxito (Si/No)

Nota:

Inicie cada objetivo desde la pantalla principal (Calendario de actividades)

Nivel de carrera:

Nota: presione (Ctrl + h) para insertar la hora

Objetivo	Nro	Descripción de la tarea	Inicio	Fin	Total	Errores	Finalizó	Observación
Interactuar con el modelo 3D y centrar la FICA en pantalla	1	Rotar, trasladar y escalar modelo 3D			00:00:00			
	2	Usar gestos de interacción para centrar el edificio de la FICA en pantalla			00:00:00			
Visualizar información del Edificio Central y	3	Seleccionar edificio Central y abrir ventana de información			00:00:00			
	4	Seleccionar edificio de Posgrado y abrir ventana de información			00:00:00			
Reiniciar la ubicación de la cámara	5	Abrir submenú en la esquina inferior izquierda y seleccionar el botón "Reiniciar"			00:00:00			
Cambiar entre modos No AR -> AR AR -> No AR	6	Abrir submenú en la esquina inferior izquierda y seleccionar "Alternar AR"			00:00:00			
	7	Abrir submenú en la esquina inferior izquierda y seleccionar "Alternar No AR"			00:00:00			
Cerrar aplicación usando UI	8	Abrir submenú en la esquina inferior izquierda y seleccionar el botón "Salir"			00:00:00			
Total					0:00:00			

Final

Una vez finalizado el taller guárdelo y complete la encuesta.

Anexo F. Tabulación de los datos obtenidos del taller práctico aplicado en la evaluación del aplicativo Paseo UTN

Nro.	tarea	n	tiempo_promedio_seg	% finalización	% errores
1	Rotar, trasladar y escalar modelo 3D	46	50.47727273	97.7	4.5
2	Usar gestos de interacción para centrar el edificio de la FICA en pantalla	46	40.13636364	97.7	0
3	Seleccionar edificio Central y abrir ventana de información	46	28.36363636	93.2	4.5
4	Seleccionar edificio de Posgrado y abrir ventana de información	46	33.22727273	93.2	6.8
5	Abrir submenú en la esquina inferior izquierda y seleccionar el botón "Reiniciar"	46	13.56818182	97.7	0
6	Abrir submenú en la esquina inferior izquierda y seleccionar "Alternar AR"	46	29.20454545	97.7	11.4
7	Abrir submenú en la esquina inferior izquierda y seleccionar "Alternar No AR"	46	21.79545455	97.7	4.5
8	Abrir submenú en la esquina inferior izquierda y seleccionar el botón "Salir"	46	17.88636364	97.7	0

encuestados	registros	% finalización global	% con error global
46	352	96.6	4

Preguntas	promedio_corr	tipo
1. ¿Considera que utilizaría esta aplicación con frecuencia?	3.368421053	positiva
10. ¿Considera que se requieren conocimientos técnicos previos para utilizar correctamente la	4	negativa
2. ¿Considera que la aplicación es compleja?	3.578947368	negativa
3. ¿Considera que la aplicación es fácil de utilizar?	3.684210526	positiva
4. ¿Considera que necesitaría apoyo de un experto para poder utilizar esta aplicación?	4.4375	negativa
5. ¿Considera que la aplicación presenta	4.375	negativa
6. ¿Considera que esta aplicación es fácil de aprender para la mayoría de las personas?	4	positiva
7. ¿Considera que la aplicación resulta confusa de utilizar?	4.066666667	negativa
8. El sistema fue muy confuso	4.533333333	negativa
9. ¿Se sintió seguro/a al utilizar la aplicación?	4.4	positiva

encuestados	respuestas_total	promedio_global_corr
46	165	4.012

Anexo G. Manual Técnico de la aplicación

MANUAL TÉCNICO

**UTN Móvil
Paseo UTN**

Número de versión: 1.0

Fecha de elaboración: 10 de febrero de 2026

Índice

1. Glosario de términos y siglas	77
1.1 Términos generales	77
1.2 Unity y conceptos base	77
1.3 Modelo 3D y contenido	78
1.4 AssetBundle y carga dinámica	78
1.5 Cifrado y seguridad	78
1.6 Escenas específicas del sistema	79
1.7 Interacción y entrada de usuario	79
1.8 Cámara y visualización	80
1.9 Street View (SV)	81
1.10 Otros términos técnicos	81
2. Resumen del sistema	81
2.1 Visión general del sistema	81
2.2 Objetivo funcional del sistema	81
2.3 Componentes principales del sistema	82
2.3.1 Aplicación cliente (Unity)	82
2.3.2 Contenido protegido (modelo 3D)	82
2.3.3 Sistema de escenas	82
2.3.4 Sistema de interacción	83
2.3.5 Interfaz de usuario (UI)	83
2.4 Modalidades de uso del sistema	83
2.4.1 Modo NoAR (visualización tradicional)	83
2.4.2 Modo AR (Realidad Aumentada)	84
2.4.3 Modo Street View (SV)	84
2.5 Flujo de uso desde la perspectiva del usuario	84

2.6 Características técnicas relevantes	85
3. Requisitos y dependencias	85
3.1 Requisitos de hardware	85
3.1.1 Equipos de desarrollo	85
3.1.2 Dispositivos móviles de prueba	85
3.2 Requisitos de software	86
3.2.1 Sistema operativo (PC)	86
3.2.2 Motor de desarrollo	86
3.2.3 Lenguaje de programación	86
3.2.4 SDKs y herramientas externas	86
3.2.5 Paquetes y frameworks Unity	86
3.3 Requisitos de red y conectividad	87
3.4 Requisitos de permisos en dispositivos móviles	87
3.4.1 Android.....	87
3.4.2 iOS.....	87
3.5 Requisitos de acceso y roles	87
3.6 Riesgos asociados a dependencias	87
4. Estructura del proyecto	88
5. Arquitectura General	90
5.1 Componentes principales	91
5.2 Diagrama de flujo	93
5.3 Flujo por escenas	94
6. Seguridad del contenido	95
6.1 Objetivos de la protección	95
6.2 AssetBundle como contenedor	96
7. Pipeline de Contenido (Editor)	96
7.1 Preparación del Prefab del modelo 3D	97

7.2 Generación del AssetBundle	97
7.3 Cifrado AES del AssetBundle	97
7.4 Artefactos generados	98
8. Pipeline de ejecución (Runtime)	99
9. Escenas del sistema	99
9.1 Escena Loading	99
9.2 Escena NoAR	100
9.3 Escena AR	100
9.4 Escena SV (Street View)	100
10. Detalle por escena y scripts	100
10.1 Escena Loading	100
10.1.1 Script: AssetBundleLoader	101
10.1.2 Script: LoadingScreen	102
10.2 Escena NoAR	102
10.2.1 Script: ModelRestorer	103
10.2.2 Script: UIManager	104
10.2.3 Script: CameraController	105
10.2.4 Script: InteractionManager	106
10.3 Escena AR	106
10.3.1 Script: ModelRestorer	107
10.3.2 Script: UIManager	107
10.3.3 Script: ARSupport	108
10.3.4 Script: LocationController	108
10.3.5 Script: ShareImage	109
10.4 Escena SV (Street View)	109
10.4.1 Script: SVLoader	110
10.4.2 Script: SVSMSSelector	110

10.4.3 Script: StandManager.....	111
10.4.4 Script: SVMMap.....	112
10.4.5 Script: Stand.....	112
10.4.6 Script: Node.....	112
10.4.7 Script: Sign.....	113
11. Interacción del usuario	113
11.1 Gestos en NoAR (cámara)	113
11.2 Gestos en AR (manipulación del modelo)	114
11.3 Selección de edificios (tap / doble tap)	114
11.4 Interacción en Street View (SV)	114
12. Realidad Aumentada	115
12.1 Detección de compatibilidad (ARSupport)	115
12.2 Comportamiento en dispositivos no compatibles	115
13. Street View 360	116
14. Procedimientos de mantenimiento	117
14.1 Actualizar el modelo 3D	117
14.2 Regenerar el AssetBundle	119
14.3 Re-cifrar y reemplazar el archivo .enc	120
14.4 Probar antes de liberar	121
15. Compilación y despliegue	121
15.1 Escena inicial obligatoria (Loading)	122
15.2 Configuraciones técnicas por plataforma	122
15.3 Proceso de build (ambas plataformas)	123
15.4 Firma y despliegue en Android	123
Generación del archivo .aab.....	123
15.5 Firma y despliegue en iOS	125
16. Evaluación del rendimiento del desarrollo	126

16.1 Métricas evaluadas	126
16.1.1 Tiempo de compilación	126
16.1.2 Tiempo de carga de escenas	126
16.1.3 Uso de recursos	127
16.1.4 Tamaño del proyecto y compilaciones.....	127
16.1.5 Escalabilidad del contenido.....	127
16.1.6 Mantenibilidad del código.....	127
Consideraciones a futuro	128
Conexión con una base de datos pública sin datos sensibles	128
Implementación del módulo de búsqueda desarrollado	128
Activación de la funcionalidad de alternar calidad gráfica	128
Desarrollo de módulos para otros campus	128
Vinculación de lugares internos con la base de datos para guía y navegación	129

1. Glosario de términos y siglas

El presente glosario define los términos técnicos utilizados a lo largo de este manual. Su objetivo es facilitar la comprensión del funcionamiento de la aplicación **Paseo UTN** al personal del Departamento de Tecnología, incluso si no posee experiencia previa en Unity, desarrollo de videojuegos o realidad aumentada.

1.1 Términos generales

Aplicación móvil

Software diseñado para ejecutarse en dispositivos móviles (Android e iOS), instalado localmente en el dispositivo del usuario.

Offline

Funcionamiento de la aplicación sin necesidad de conexión a Internet. La aplicación Paseo UTN puede operar completamente offline una vez instalada.

Runtime

Momento en el que la aplicación ya está instalada y se encuentra en ejecución en el dispositivo del usuario.

Editor (Unity Editor)

Entorno de desarrollo utilizado para crear, configurar y compilar la aplicación. No forma parte de la aplicación final instalada en el dispositivo.

1.2 Unity y conceptos base

Unity

Motor de desarrollo multiplataforma utilizado para crear la aplicación. Gestiona gráficos 3D, escenas, interacción, lógica y compilación a Android e iOS.

Escena (Scene)

Unidad principal de organización en Unity. Cada escena representa un estado o pantalla completa de la aplicación (por ejemplo: Loading, NoAR, AR, SV).

GameObject (GO)

Entidad base dentro de una escena de Unity. Puede representar un objeto visual, un controlador lógico, una cámara, un botón, etc.

Componente (Component)

Bloque funcional que se adjunta a un GameObject para darle comportamiento (por ejemplo: scripts, colliders, renderers).

Script

Archivo de código (en C#) que define el comportamiento de uno o varios GameObjects.

En este proyecto, los scripts controlan carga de contenido, interacción, UI, AR y Street View.

1.3 Modelo 3D y contenido

Modelo 3D

Representación tridimensional del campus universitario, incluyendo edificios, espacios y elementos relevantes.

Prefab

Plantilla de Unity que agrupa un objeto (o conjunto de objetos) con sus configuraciones, componentes y scripts listos para ser instanciados en una escena.

Instanciar

Crear una copia activa de un objeto (Prefab) dentro de una escena durante la ejecución de la aplicación.

1.4 AssetBundle y carga dinámica

AssetBundle

Archivo binario propio de Unity que contiene recursos (modelos 3D, texturas, materiales, prefabs). Permite cargar contenido de forma dinámica en tiempo de ejecución.

Carga dinámica

Proceso mediante el cual el contenido no se incluye directamente en la escena, sino que se carga en memoria cuando la aplicación está en ejecución.

StreamingAssets

Carpeta especial de Unity cuyo contenido se incluye en la aplicación final sin modificaciones y puede ser leído directamente en runtime.

1.5 Cifrado y seguridad

Cifrado

Proceso de transformación de datos para que no puedan ser leídos sin la clave correcta.

AES (Advanced Encryption Standard)

Algoritmo criptográfico simétrico utilizado para cifrar el AssetBundle del modelo 3D.

CBC (Cipher Block Chaining)

Modo de operación del algoritmo AES que encadena bloques para aumentar la seguridad del cifrado.

PKCS7

Esquema de padding utilizado para completar bloques de datos en el cifrado AES.

Clave (Key)

Valor secreto utilizado por el algoritmo AES para cifrar y descifrar el contenido. En esta aplicación:

- Está escrita en el código del script de cifrado (Editor).
- Está ofuscada en el código del script de descifrado (Runtime).

IV (Initialization Vector)

Valor inicial utilizado por AES en modo CBC para garantizar que el cifrado no produzca patrones repetitivos.

Ofuscación

Técnica que dificulta la lectura directa del código o valores sensibles (como la clave), sin ocultarlos completamente.

1.6 Escenas específicas del sistema

Loading

Escena inicial de la aplicación. Se encarga de cargar y descifrar el contenido antes de permitir el acceso al resto de la app.

NoAR

Escena de visualización del modelo 3D sin realidad aumentada. Funciona como una galería 3D interactiva.

AR (Realidad Aumentada)

Escena donde el modelo 3D se muestra sobre el entorno real captado por la cámara del dispositivo.

SV (Street View)

Escena que muestra fotografías 360° y permite navegación entre puntos (stands) mediante nodos interactivos.

1.7 Interacción y entrada de usuario

Gestos táctiles

Interacciones realizadas por el usuario mediante uno o más dedos sobre la pantalla (tap, doble tap, arrastre, pinch).

Tap

Toque simple sobre la pantalla.

Doble tap

Dos toques consecutivos utilizados para acciones especiales (por ejemplo, centrar la cámara en un edificio).

Raycast

Técnica que lanza un “rayo invisible” desde la cámara para detectar qué objeto está siendo tocado por el usuario.

1.8 Cámara y visualización

Camera.main

Cámara principal de la escena. Muchos scripts dependen de su existencia para funcionar correctamente.

Pivot

Punto de referencia usado para rotar y escalar el modelo 3D de forma estable.

LookAtConstraint

Componente que fuerza a un objeto a mirar siempre hacia otro (por ejemplo, un cartel mirando a la cámara).

3.9 Interfaz de usuario (UI)

UI (User Interface)

Conjunto de elementos visuales con los que interactúa el usuario: botones, paneles, menús, textos.

Canvas

Contenedor principal de la interfaz gráfica en Unity.

Portrait / Landscape

Orientaciones vertical y horizontal del dispositivo móvil.

3.10 Realidad Aumentada

AR Foundation

Framework de Unity que unifica el uso de ARCore (Android) y ARKit (iOS).

ARCore

Plataforma de realidad aumentada de Android.

ARKit

Plataforma de realidad aumentada de iOS.

Compatibilidad AR

Capacidad del dispositivo para ejecutar funciones de realidad aumentada. No todos los dispositivos son compatibles.

1.9 Street View (SV)

Stand

Punto o estación dentro del modo Street View que contiene una imagen 360° y su configuración.

Nodo (Node)

Objeto interactivo que permite moverse de un stand a otro.

Imagen 360°

Fotografía esférica que permite visualizar el entorno en todas las direcciones.

1.10 Otros términos técnicos

Singleton

Patrón de diseño que garantiza que exista una sola instancia de un script durante la ejecución.

Build

Proceso de compilación de la aplicación para generar el instalador (APK en Android, IPA en iOS).

Inspector

Panel del Editor de Unity donde se configuran valores y referencias de los scripts.

2. Resumen del sistema

2.1 Visión general del sistema

La aplicación Paseo UTN es un sistema móvil interactivo diseñado para la visualización, exploración e interacción con un modelo 3D del campus universitario, complementado con funcionalidades de Realidad Aumentada (AR) y un modo de navegación tipo Street View basado en imágenes 360°.

El sistema ha sido concebido como una aplicación autónoma, offline y modular, donde el contenido principal (modelo 3D) se encuentra protegido mediante cifrado y se carga dinámicamente en tiempo de ejecución.

Desde el punto de vista técnico, la aplicación no es una app tradicional de formularios o pantallas estáticas, sino un sistema gráfico interactivo en tiempo real.

2.2 Objetivo funcional del sistema

El sistema tiene como objetivos principales:

- Permitir la **exploración visual del campus universitario** sin necesidad de presencia física.

- Facilitar la **identificación de edificios y espacios** mediante nombres, carteles e información asociada.
- Ofrecer múltiples formas de visualización según el dispositivo:
 - Visualización 3D tradicional (NoAR).
 - Visualización en Realidad Aumentada (AR).
 - Navegación inmersiva mediante imágenes 360° (Street View).
- Proteger el modelo 3D institucional contra extracción o uso no autorizado.

2.3 Componentes principales del sistema

El sistema está compuesto por los siguientes bloques funcionales de alto nivel:

2.3.1 Aplicación cliente (Unity)

Es la aplicación instalada en el dispositivo móvil del usuario. Se encarga de:

- Ejecutar la lógica de la aplicación.
- Renderizar el modelo 3D.
- Gestionar la interacción del usuario.
- Controlar la interfaz gráfica.
- Cargar y descifrar el contenido protegido.

No existe backend activo ni dependencia de servidores externos para el funcionamiento básico.

2.3.2 Contenido protegido (modelo 3D)

El contenido principal del sistema es el **modelo 3D del campus**, que incluye:

- Edificios principales.
- Espacios secundarios.
- Elementos visuales de referencia.

Este contenido:

- Se empaqueta como AssetBundle.
- Se cifra antes de incluirse en la aplicación.
- Se descifra solo en memoria durante la ejecución.

2.3.3 Sistema de escenas

La aplicación se organiza en **escenas independientes**, cada una con una responsabilidad clara:

- **Loading:** carga y descifrado del contenido.
- **NoAR:** visualización 3D tradicional.
- **AR:** visualización del modelo en realidad aumentada.
- **SV:** visualización Street View 360°.

Cada escena tiene scripts específicos y no comparte lógica innecesaria con otras escenas.

2.3.4 Sistema de interacción

El sistema permite la interacción mediante:

- Gestos táctiles (uno o dos dedos).
- Toques simples y dobles.
- Selección directa de objetos del modelo 3D.
- Navegación por nodos en Street View.

Toda la interacción está gestionada mediante scripts dedicados y no depende de interfaces externas.

2.3.5 Interfaz de usuario (UI)

La UI es responsable de:

- Navegación entre escenas.
- Visualización de información contextual.
- Adaptación automática a orientación vertical u horizontal.
- Control de acceso a AR y Street View.

La UI no contiene lógica de negocio crítica, pero es esencial para el uso del sistema.

2.4 Modalidades de uso del sistema

El sistema ofrece **tres modos principales de operación**, que comparten el mismo contenido base, pero difieren en su presentación e interacción.

2.4.1 Modo NoAR (visualización tradicional)

En este modo:

- El modelo 3D se muestra en una escena 3D controlada por cámara.
- El usuario puede:
 - Mover la cámara.
 - Rotar y acercarse al modelo.

- Seleccionar edificios.
- No se requiere hardware compatible con AR.

Este modo garantiza que la aplicación sea usable en dispositivos no compatibles con realidad aumentada.

2.4.2 Modo AR (Realidad Aumentada)

En este modo:

- El modelo 3D se superpone al entorno real captado por la cámara.
- El usuario puede:
 - Colocar el modelo en el espacio.
 - Escalarlo y rotarlo.
 - Visualizar el campus en su entorno físico.
- El sistema verifica previamente la compatibilidad del dispositivo.

Este modo depende de AR Foundation y del soporte nativo del dispositivo.

2.4.3 Modo Street View (SV)

En este modo:

- El usuario navega mediante imágenes 360°.
- Cada punto (stand) representa una ubicación específica.
- La navegación se realiza seleccionando nodos (esferas rojas) interactivos.

Este modo permite una experiencia inmersiva complementaria al modelo 3D.

2.5 Flujo de uso desde la perspectiva del usuario

De forma simplificada, el uso típico del sistema sigue el siguiente flujo:

1. El usuario abre la aplicación.
2. Se muestra una pantalla de carga.
3. El contenido se descifra y carga.
4. El usuario accede al modo NoAR o AR.
5. Interactúa con el modelo y selecciona edificios.
6. Opcionalmente, accede al modo Street View.
7. Regresa al modo principal o cierra la aplicación.

Este flujo está diseñado para ser lineal, predecible y sin dependencias externas.

2.6 Características técnicas relevantes

Desde el punto de vista técnico, el sistema presenta las siguientes características clave:

- Funcionamiento completamente offline.
- Protección del contenido mediante cifrado AES.
- Carga dinámica de contenido.
- Separación clara de responsabilidades por escena.
- Adaptación automática a distintos dispositivos.
- Uso intensivo de gráficos 3D y renderizado en tiempo real.

3. Requisitos y dependencias

Esta sección detalla todos los requisitos técnicos, dependencias de software, hardware y accesos necesarios para poder instalar, mantener, modificar, compilar y desplegar la aplicación Paseo UTN.

3.1 Requisitos de hardware

3.1.1 Equipos de desarrollo

Para abrir, compilar y mantener el proyecto Unity se requiere al menos:

- **Procesador:**
CPU de 4 núcleos o superior (Intel i5 / AMD Ryzen 5 o equivalente).
- **Memoria RAM:**
Mínimo 16 GB (recomendado 32 GB para escenas 3D pesadas).
- **Almacenamiento:**
Al menos 50 GB libres (el proyecto incluye modelos 3D de gran tamaño, AssetBundles y builds temporales).
- **GPU:**
Tarjeta gráfica dedicada o integrada moderna con soporte para gráficos 3D (OpenGL / Vulkan / Metal).

3.1.2 Dispositivos móviles de prueba

Android

- Versión mínima recomendada: **Android 7.0 (API 24)** o superior.
- Soporte AR (opcional): dispositivos compatibles con ARCore.
- Almacenamiento disponible: al menos 1 GB libre.

iOS

- Versión mínima: **iOS 13**.
- Dispositivos compatibles con ARKit para modo AR.
- Espacio libre suficiente para la instalación del IPA.

3.2 Requisitos de software

3.2.1 Sistema operativo (PC)

Cualquiera de los siguientes:

- Windows 10 / 11 (64 bits).
- macOS (requerido para compilación iOS).
- Linux (solo para desarrollo parcial; no permite build iOS).

3.2.2 Motor de desarrollo

- **Unity Hub** (gestor de versiones).
- **Unity Editor** (versión exacta utilizada en el proyecto (recomendado mantenerla sin cambios para evitar incompatibilidades).

Nota: cambiar la versión de Unity sin pruebas puede romper AssetBundles, AR o shaders.

3.2.3 Lenguaje de programación

- **C#** (gestionado internamente por Unity).
- No se requiere instalar compiladores externos.

3.2.4 SDKs y herramientas externas

Android

- Android SDK (instalado desde Unity Hub).
- Android NDK (según configuración del proyecto).
- OpenJDK incluido con Unity.

iOS

- **Xcode** (versión compatible con la versión de Unity).
- Cuenta de desarrollador Apple (para firma y distribución).

3.2.5 Paquetes y frameworks Unity

- AR Foundation.
- ARCore XR Plugin (Android).
- ARKit XR Plugin (iOS).

- Addressables (para Street View en Android).
- Unity UI (UGUI).

Todos estos paquetes se gestionan desde el **Package Manager** de Unity.

3.3 Requisitos de red y conectividad

- **No se requiere conexión a Internet** para el funcionamiento normal de la aplicación.
- La conexión solo es necesaria para:
 - Descargar Unity.
 - Instalar paquetes.
 - Subir builds a tiendas (Google Play / App Store).

3.4 Requisitos de permisos en dispositivos móviles

3.4.1 Android

- Acceso a cámara (modo AR).
- Permisos de compartición (ShareImage).

3.4.2 iOS

- Acceso a cámara (modo AR).
- Permisos estándar definidos en Info.plist.

3.5 Requisitos de acceso y roles

Para trabajar correctamente con el proyecto se requiere:

- Acceso al repositorio del proyecto Unity.
- Acceso a los archivos del modelo 3D original.
- Acceso al script de cifrado (Encryptor AES).
- Conocimiento de la **clave de cifrado** utilizada (para regenerar contenido).
- Acceso a cuentas de desarrollador (Google / Apple) para despliegue.

3.6 Riesgos asociados a dependencias

- Cambios de versión de Unity pueden romper compatibilidad.
- Cambios en SDKs móviles pueden afectar AR.
- Pérdida de la clave de cifrado impide regenerar contenido.

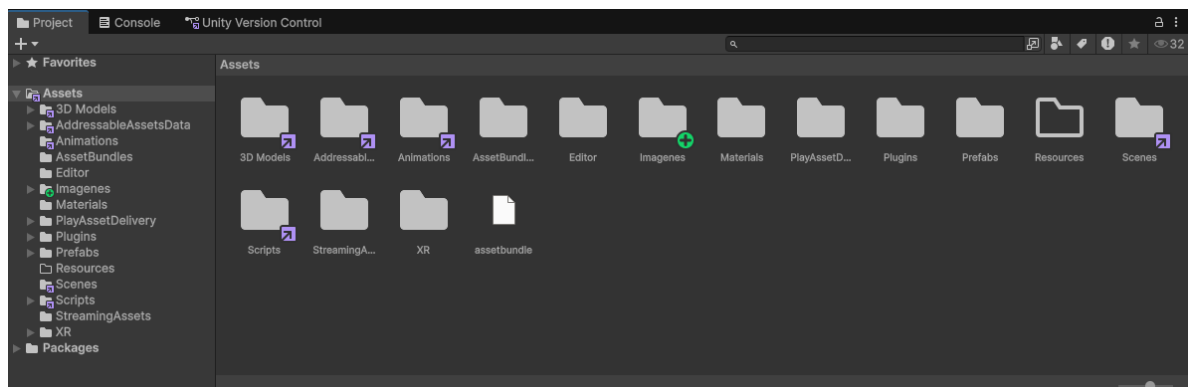
- Eliminación o corrupción del .enc inutiliza la app.

El incumplimiento de alguno de estos requisitos puede provocar fallos críticos en la aplicación.

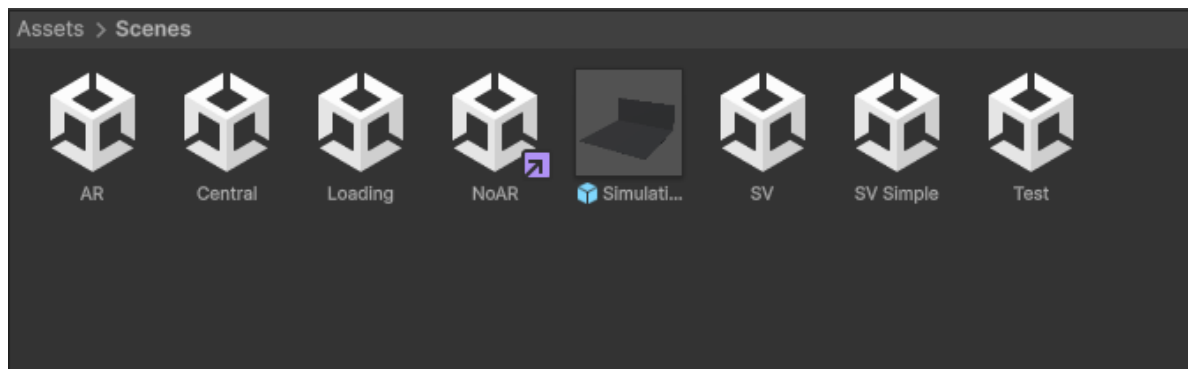
4. Estructura del proyecto

El proyecto de la aplicación Paseo UTN está desarrollado utilizando el motor Unity y se organiza siguiendo la estructura estándar de este entorno. La finalidad de esta organización es separar claramente los recursos visuales, la lógica de funcionamiento y el contenido protegido de la aplicación, de forma que el mantenimiento y la actualización del proyecto puedan realizarse de manera ordenada y controlada.

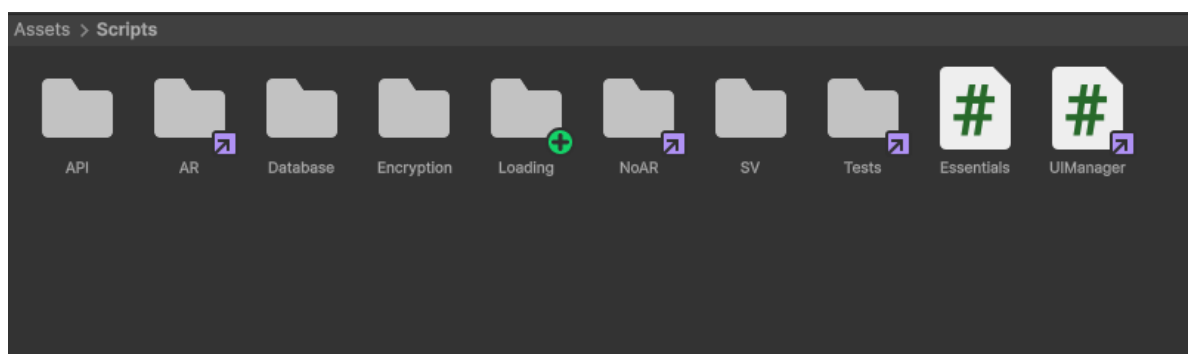
Todo el contenido relevante del proyecto se encuentra dentro de la carpeta principal denominada Assets. Esta carpeta actúa como contenedor central de escenas, scripts, modelos, recursos gráficos y archivos necesarios para la ejecución de la aplicación. Cualquier modificación realizada dentro de esta carpeta impacta directamente en el funcionamiento del sistema, por lo que se recomienda realizar cambios únicamente cuando sea necesario y siempre recompilar la aplicación después de hacerlo.



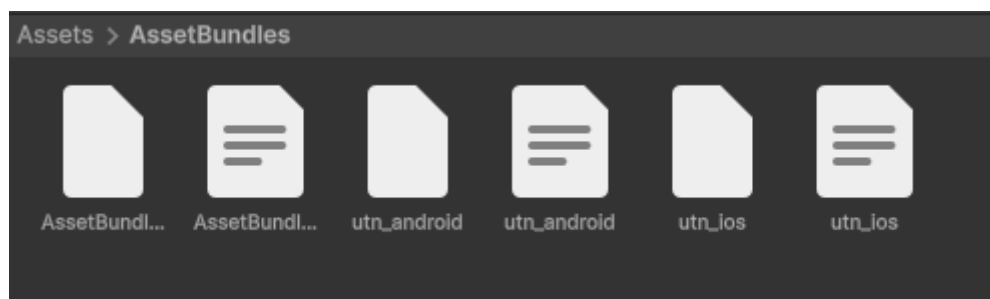
Dentro del proyecto existen varias carpetas cuyo propósito es claramente diferenciado. Las escenas de la aplicación se almacenan en una carpeta dedicada a este fin, donde cada escena representa un estado específico del sistema, como la pantalla de carga, la visualización del modelo sin realidad aumentada, el modo de realidad aumentada y el modo Street View. Estas escenas no son intercambiables y cumplen roles definidos dentro del flujo general de la aplicación.



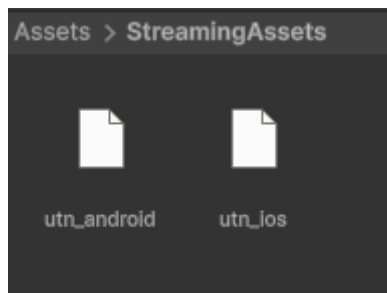
La lógica de funcionamiento se encuentra agrupada en scripts escritos en C#. Estos scripts controlan aspectos como la carga del contenido, la interacción del usuario, la interfaz gráfica, la detección de compatibilidad con realidad aumentada y la navegación entre escenas. Aunque el proyecto contiene múltiples scripts, su organización responde a la necesidad de mantener separadas las responsabilidades de cada módulo, lo que facilita la identificación de errores y el mantenimiento futuro.



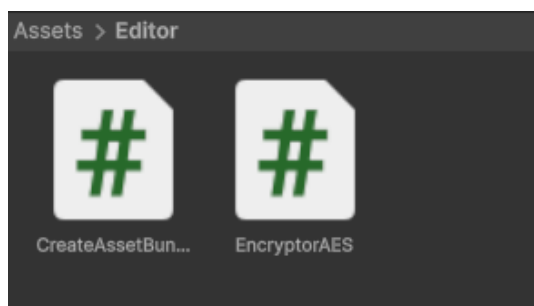
El modelo 3D principal del campus no se encuentra directamente integrado dentro de las escenas. En su lugar, se maneja como un recurso externo empaquetado en un AssetBundle, lo que permite protegerlo y cargarlo dinámicamente en tiempo de ejecución. Los archivos relacionados con este proceso se almacenan en ubicaciones específicas del proyecto y no deben ser modificados manualmente.



Una carpeta de especial importancia es StreamingAssets, ya que contiene los archivos cifrados que permiten a la aplicación cargar el modelo 3D al iniciar. Estos archivos son incluidos en la aplicación final sin ser modificados por Unity y son leídos directamente durante la ejecución. La ausencia o corrupción de estos archivos provoca que la aplicación no pueda completar su proceso de carga inicial, quedando inutilizable.



El proyecto también incluye scripts que solo se utilizan durante la fase de desarrollo y no forman parte de la aplicación final instalada en los dispositivos. Estos scripts se encargan de tareas como la generación del AssetBundle y el cifrado del contenido. Aunque son fundamentales para la actualización del modelo 3D, no influyen en el comportamiento de la aplicación una vez desplegada.

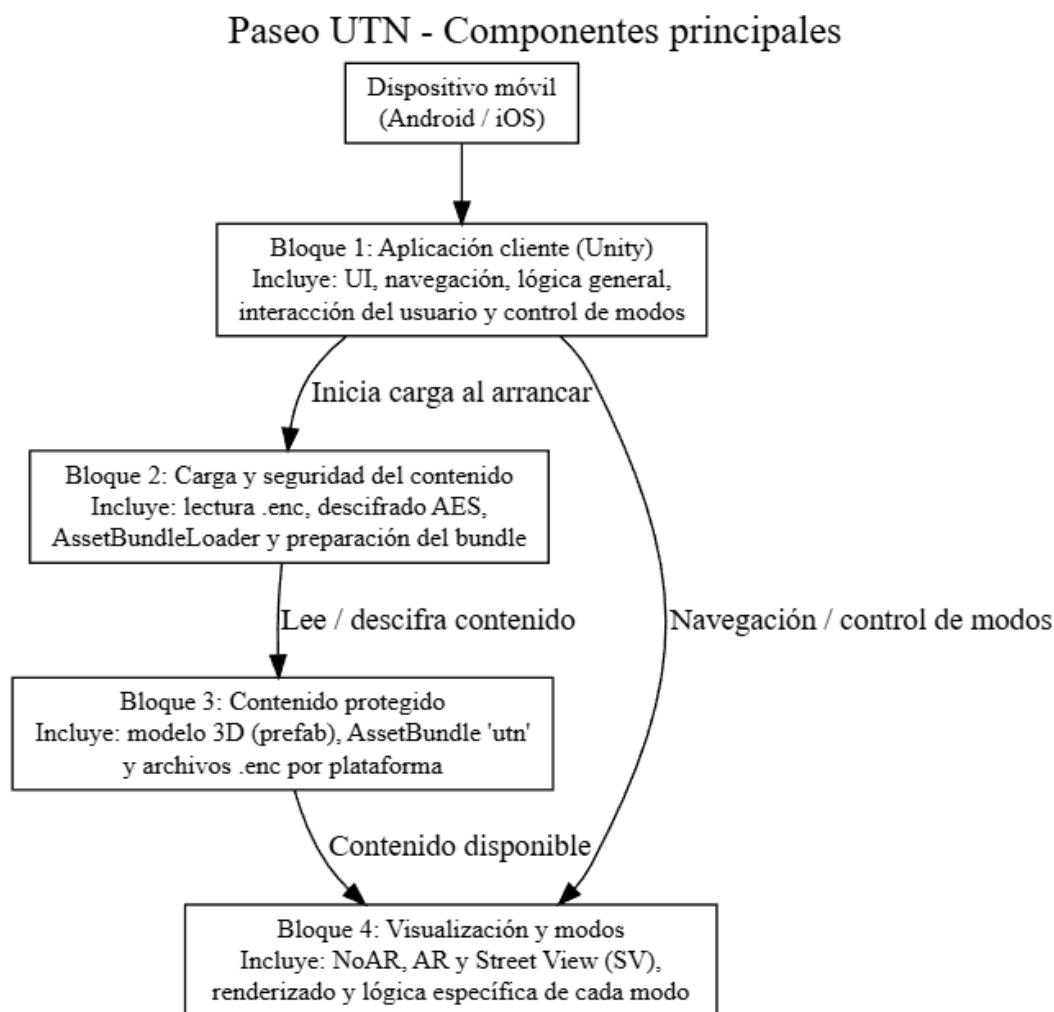


Este apartado no pretende describir el funcionamiento interno de cada componente, sino ofrecer una visión general que permita entender cómo se organiza el proyecto y cuáles son las áreas que requieren mayor cuidado durante el mantenimiento.

5. Arquitectura General

La arquitectura de la aplicación Paseo UTN define cómo se organizan y relacionan los distintos componentes del sistema para permitir la carga segura del contenido, la visualización del modelo 3D y la interacción del usuario. Este apartado tiene como objetivo que el personal del DDTI pueda comprender el funcionamiento global de la aplicación sin necesidad de conocer detalles internos del motor Unity.

5.1 Componentes principales



El primer bloque corresponde a la aplicación cliente, desarrollada en Unity y ejecutada directamente en el dispositivo móvil. Este bloque se encarga de gestionar la ejecución del sistema, el renderizado gráfico, la interfaz de usuario y la interacción con el usuario. No existe un servidor externo ni dependencias en línea para el funcionamiento normal de la aplicación.

El segundo bloque está constituido por el contenido protegido, que corresponde al modelo 3D del campus universitario. Este contenido no se encuentra integrado directamente en las escenas, sino que se distribuye como un archivo cifrado que se carga dinámicamente en tiempo de ejecución. Esta decisión arquitectónica responde a la necesidad de proteger el modelo institucional y controlar su distribución.

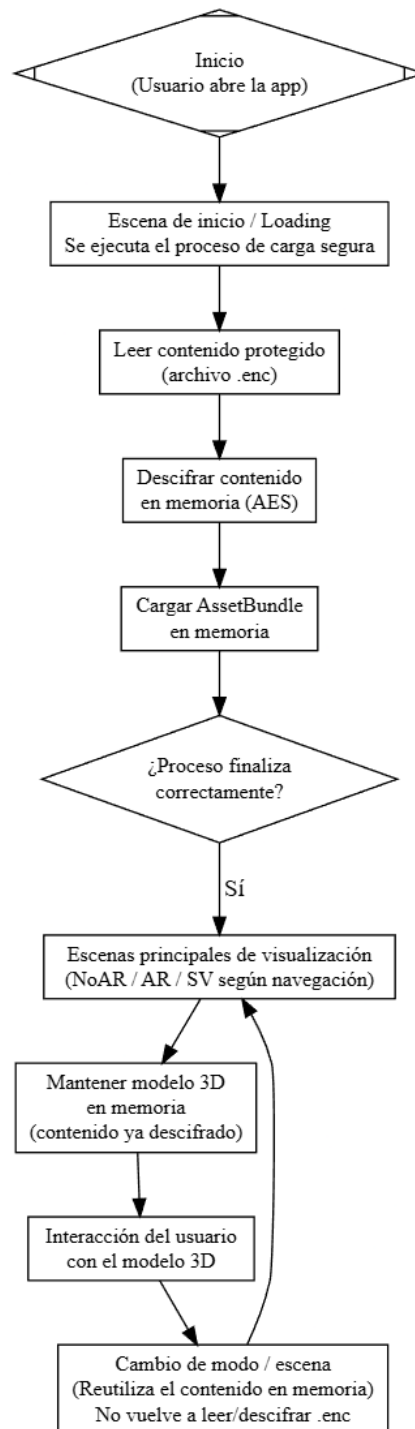
El tercer bloque corresponde al sistema de carga y seguridad. Este bloque es responsable de leer el archivo cifrado, descifrarlo en memoria y dejar disponible el

contenido para el resto de la aplicación. Este proceso ocurre al inicio de la aplicación y es un paso obligatorio antes de permitir cualquier interacción.

El cuarto bloque es el sistema de visualización e interacción, que engloba las escenas NoAR, AR y Street View, junto con la lógica que permite al usuario navegar entre ellas, interactuar con el modelo 3D, seleccionar edificios y acceder a información adicional.

5.2 Diagrama de flujo

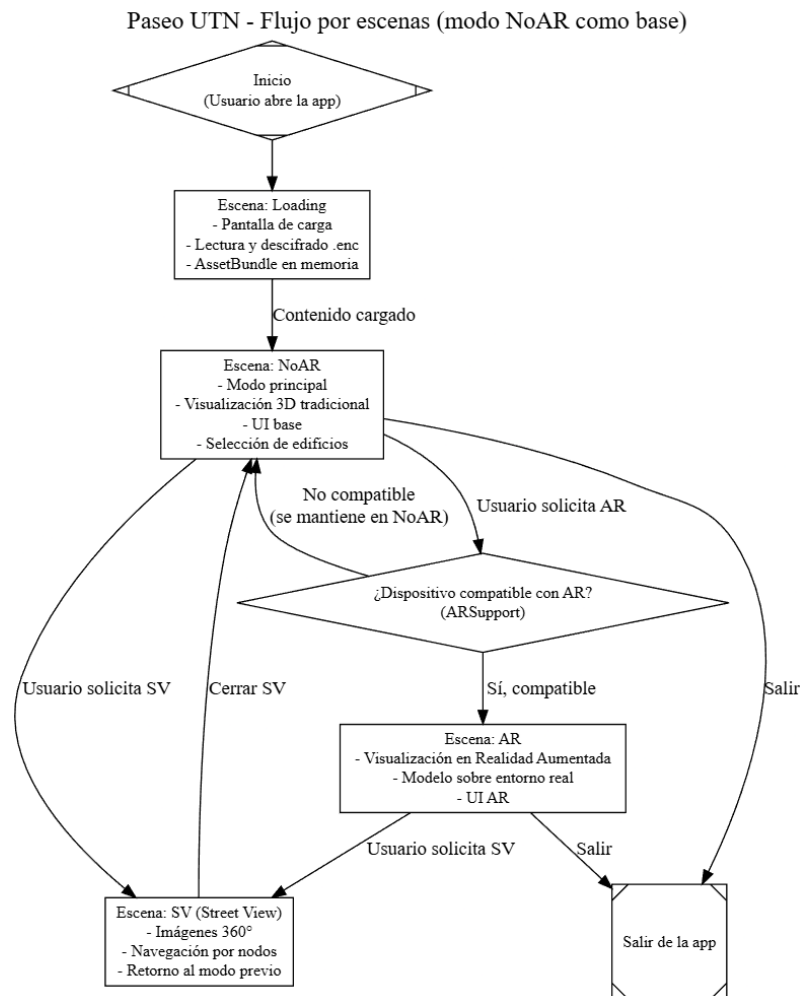
Paseo UTN - Flujo de ejecución (lectura/descifrado y reutilización en memoria)



Al iniciar la aplicación, se carga inmediatamente la escena de inicio o carga. En esta escena se ejecuta el proceso de lectura y descifrado del contenido protegido. Solo cuando este proceso finaliza correctamente, la aplicación permite el paso a las escenas principales de visualización.

Una vez cargado el contenido, el sistema mantiene el modelo 3D en memoria y permite al usuario interactuar con él a través de las diferentes escenas. El cambio entre modos de visualización no implica volver a cargar el contenido cifrado, sino reutilizar el contenido ya descifrado, lo que optimiza el rendimiento y reduce tiempos de espera.

5.3 Flujo por escenas



La escena de carga actúa como punto de entrada obligatorio al sistema. Desde esta escena, una vez que el contenido está disponible, la aplicación transiciona a la escena principal de visualización sin realidad aumentada o, según la configuración y compatibilidad del dispositivo, a la escena de realidad aumentada.

Desde las escenas NoAR o AR, el usuario puede acceder al modo Street View. Este modo se carga de forma independiente y, al cerrarse, retorna a la escena desde la cual fue invocado. Este comportamiento garantiza que el usuario no pierda el contexto de navegación ni el estado del modelo seleccionado.

6. Seguridad del contenido

La aplicación Paseo UTN maneja como elemento principal un modelo 3D institucional del campus. Debido al valor del contenido y a la facilidad con la que un archivo 3D puede copiarse si se distribuye sin protección, el proyecto incorpora un mecanismo de seguridad enfocado en evitar que el modelo pueda ser extraído de forma directa desde la aplicación instalada.

El enfoque de seguridad aplicado no se basa en autenticación de usuarios ni en servidores externos. En su lugar, se protege el recurso más sensible del sistema, que es el modelo 3D, mediante empaquetado y cifrado local, manteniendo el funcionamiento completamente offline.

6.1 Objetivos de la protección

El objetivo principal es impedir, o al menos dificultar significativamente, que el modelo 3D del campus pueda ser obtenido en un formato reutilizable a partir del instalador o del almacenamiento interno del dispositivo.

En términos operativos, esto significa que el modelo no debe existir dentro de la aplicación como un archivo accesible y legible (por ejemplo, un prefab o malla 3D incluida directamente en las escenas o dentro del paquete en un formato reconocible). En vez de ello, el contenido se distribuye en un formato que solo la aplicación puede interpretar durante la ejecución.

Este mecanismo busca proteger el modelo contra escenarios comunes como:

- Extracción del contenido desde el APK/IPA,
- Copia directa de archivos internos de la app,
- Inspección de recursos incluidos en el paquete de instalación.

El sistema no pretende afirmar que la extracción sea imposible en todos los escenarios, sino establecer una barrera técnica razonable para uso institucional, considerando que la aplicación debe mantenerse operativa sin conexión y sin servicios externos.

6.2 AssetBundle como contenedor

Para implementar la protección, el modelo 3D no se integra directamente en las escenas principales. En su lugar, se empaqueta como un AssetBundle estándar de Unity. Este AssetBundle funciona como un contenedor binario que puede incluir el modelo en forma de prefab, junto con sus configuraciones y scripts asociados, dejando el contenido listo para ser instanciado cuando la aplicación lo requiera.

El uso del AssetBundle cumple dos propósitos. Primero, permite separar el contenido del proyecto base, evitando que el modelo permanezca incrustado de manera directa en la aplicación. Segundo, habilita un flujo controlado de actualización, donde el contenido puede regenerarse de forma independiente, siempre que se siga el procedimiento establecido.

Una vez generado el AssetBundle, este no se distribuye tal cual. Antes de integrarlo en la aplicación final, el archivo se cifra utilizando AES. Como resultado, la aplicación no incluye un AssetBundle legible, sino un archivo cifrado con extensión .enc ubicado en StreamingAssets.

Durante la ejecución, el sistema realiza el proceso inverso: lee el archivo cifrado, lo descifra en memoria y recién entonces carga el AssetBundle para instanciar el modelo. Esto implica que el contenido solo existe en un estado utilizable dentro de la memoria de la aplicación mientras está en ejecución, y no como un archivo disponible en texto plano dentro del paquete de instalación.

Desde el punto de vista de mantenimiento institucional, este diseño obliga a que cualquier actualización del modelo 3D se realice a través del pipeline oficial de regeneración y cifrado, ya que reemplazar manualmente archivos sin seguir ese proceso provoca fallos de carga o incompatibilidades.

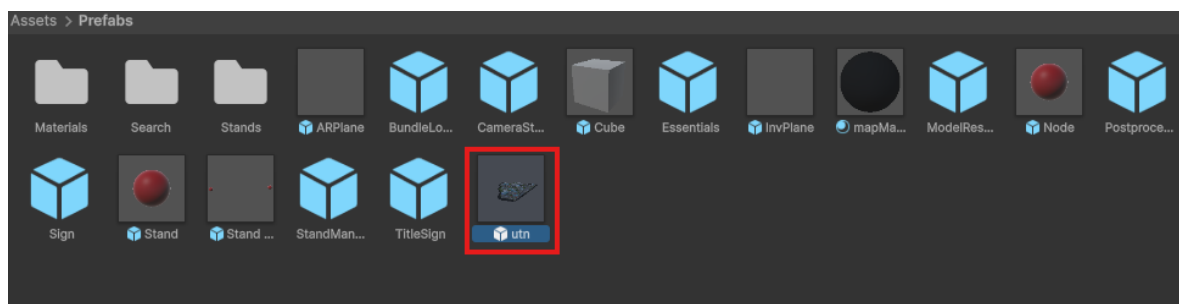
7. Pipeline de Contenido (Editor)

Este apartado describe el proceso exacto que debe seguir el DDTI para preparar, proteger y generar el contenido 3D que será utilizado por la aplicación. Todo este pipeline se ejecuta únicamente en el Editor de Unity, no en la aplicación final instalada en los dispositivos.

El objetivo de este proceso es producir un archivo cifrado .enc válido, compatible con la aplicación, que contenga el modelo 3D actualizado del campus.

7.1 Preparación del Prefab del modelo 3D

El modelo 3D del campus debe existir como un **Prefab único** (ubicado en Assets/Prefabs/utn), correctamente configurado antes de generar el AssetBundle. Este prefab es el punto de partida de todo el pipeline.



7.2 Generación del AssetBundle

Una vez validado el prefab, se utiliza el script de generación de AssetBundles (CreateAssetBundles) desde el Editor de Unity.

Este proceso:

- Toma el prefab configurado.
- Lo empaqueta en un AssetBundle por plataforma.
- Genera un archivo binario que Unity puede cargar dinámicamente.

El AssetBundle generado **no debe distribuirse directamente**, ya que aún no está protegido.

7.3 Cifrado AES del AssetBundle

El AssetBundle generado se pasa por el proceso de cifrado utilizando el script de cifrado AES (Encryptor AES).

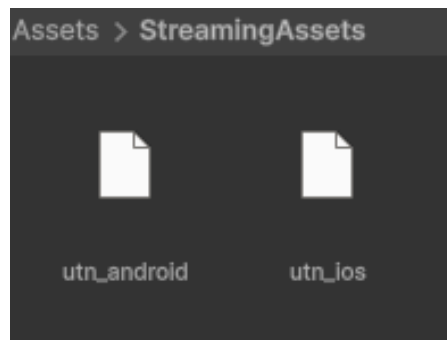
Este paso:

- Toma el AssetBundle como entrada.

- Aplica cifrado AES con la clave definida en el código de cifrado.
- Genera un archivo cifrado con extensión .enc.

La clave de cifrado utilizada en esta etapa **debe coincidir exactamente** con la lógica de descifrado utilizada en la aplicación, caso contrario el contenido no podrá cargarse en runtime.

7.4 Artefactos generados



Al finalizar el pipeline, el resultado esperado es:

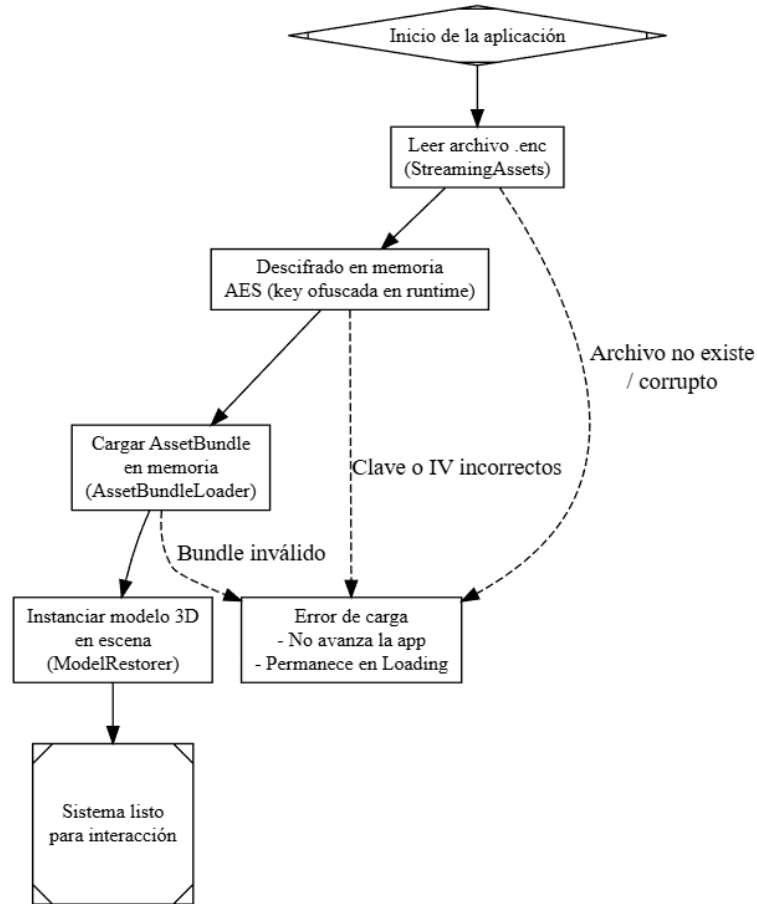
- Un archivo .enc por plataforma.
- Ningún AssetBundle en texto plano distribuido junto a la app.

El archivo .enc es el único artefacto que debe copiarse a la carpeta StreamingAssets del proyecto antes de compilar la aplicación.

No se deben modificar manualmente estos archivos ni renombrarlos, ya que la aplicación espera nombres y rutas específicas durante el proceso de carga.

8. Pipeline de ejecución (Runtime)

Paseo UTN - Pipeline de ejecución (Runtime)



Este apartado describe lo que hace la aplicación ya instalada cuando arranca. Aquí no se genera nada: solo se lee el archivo cifrado, se descifra en memoria y se habilita el modelo 3D para uso en las escenas.

9. Escenas del sistema

La aplicación está organizada por escenas. Cada escena cumple una función específica y se usa en un orden controlado; no se recomienda cambiar nombres, eliminar escenas o alterar el flujo sin pruebas completas.

9.1 Escena Loading

Es el punto de entrada técnico. Aquí se ejecuta la carga del contenido protegido: lectura del archivo .enc, descifrado y carga del AssetBundle en memoria. Si este paso no termina correctamente, la aplicación no debe avanzar a las escenas principales.

Salida esperada: cuando el contenido ya está listo, la aplicación continúa a NoAR.

9.2 Escena NoAR

Es la escena principal por defecto. Presenta el modelo 3D en un entorno 3D tradicional con cámara controlable y selección de edificios. Desde aquí se accede al resto de modos (AR y Street View) según lo permita el dispositivo y el estado del sistema.

9.3 Escena AR

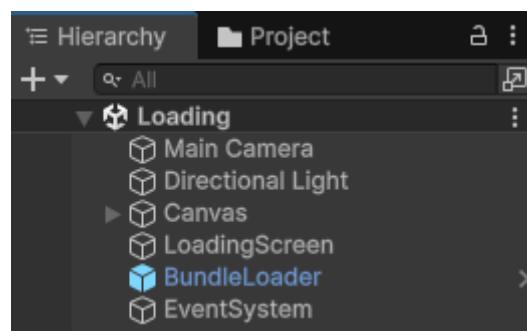
Es el modo de realidad aumentada. Solo se entra si el dispositivo es compatible. En esta escena el modelo se visualiza sobre el entorno real usando la cámara del dispositivo y se permite manipulación (rotación/zoom). Si no hay compatibilidad AR, la aplicación debe permanecer en NoAR.

9.4 Escena SV (Street View)

Es el modo de imágenes 360°. Se ingresa desde NoAR o AR. La navegación ocurre entre “stands” mediante nodos interactivos. Al cerrar SV, el comportamiento esperado es retornar a NoAR (modo base).

10. Detalle por escena y scripts

10.1 Escena Loading



La escena **Loading** es la escena técnica más crítica de toda la aplicación. Su única responsabilidad es preparar el sistema antes de permitir cualquier interacción del usuario. En esta escena no existe navegación, interacción ni visualización del modelo; todo ocurre de forma automática al iniciar la aplicación.

Si esta escena falla, la aplicación no puede continuar.

Función general de la escena

Al entrar en la escena Loading, la aplicación ejecuta el pipeline de ejecución en runtime. En términos prácticos, aquí ocurre lo siguiente:

- Se localiza el archivo cifrado que contiene el modelo 3D.
- Se descifra el contenido en memoria.
- Se carga el AssetBundle resultante.
- Se valida que el contenido esté listo para ser usado por las escenas principales.

Solo cuando este proceso termina correctamente, la aplicación puede pasar al modo NoAR.

10.1.1 Script: AssetBundleLoader

Este script es el núcleo técnico de la escena Loading. Es responsable de toda la lógica relacionada con la lectura, descifrado y carga del contenido protegido.

Su ejecución comienza automáticamente al iniciar la escena. El script busca el archivo .enc correspondiente a la plataforma desde la carpeta StreamingAssets. Una vez leído el archivo, procede a descifrarlo en memoria utilizando AES, reconstruyendo la clave ofuscada incluida en el código de la aplicación.

Cuando el descifrado finaliza correctamente, el script crea un AssetBundle en memoria y lo mantiene disponible para el resto de la aplicación. Este componente se conserva entre escenas para evitar que el contenido tenga que volver a cargarse o descifrarse.

```
void Awake()
{
    if (Instance != null && Instance != this)
    {
        Destroy(gameObject);
        return;
    }
    Instance = this;
    DontDestroyOnLoad(gameObject);
}
```

Desde el punto de vista operativo, este script **no debe modificarse** durante tareas de mantenimiento normales. Cualquier cambio en su comportamiento afecta directamente la capacidad de la aplicación para iniciar.

Cuando este script falla, los síntomas más comunes son:

- La aplicación se queda permanentemente en la pantalla de carga.
- El modelo 3D nunca aparece en las escenas posteriores.
- No se registra contenido válido para instanciar.

10.1.2 Script: LoadingScreen

El script LoadingScreen cumple una función de control. No realiza operaciones de carga ni descifrado, sino que supervisa el estado del sistema.

Durante la ejecución de la escena Loading, este script consulta de forma periódica si el AssetBundleLoader ya ha finalizado su trabajo y si el contenido está disponible. Una vez que se confirma este estado, el script autoriza el cambio hacia la escena NoAR.

```
void Update()
{
    if(AssetBundleLoader.Instance.BundleLoaded())
    {
        SceneManager.LoadScene(sceneIndex);
    }
}
```

En caso de que el contenido no esté listo, la escena Loading se mantiene activa y no permite avanzar. Este comportamiento es intencional y evita que la aplicación entre en estados incompletos o inestables.

Si este script falla o no detecta correctamente el estado de carga, la aplicación puede quedar bloqueada en Loading aun cuando el contenido ya esté disponible.

10.2 Escena NoAR

La escena **NoAR** es el **modo principal y base** de la aplicación. Toda ejecución normal del sistema desemboca en esta escena una vez que la carga inicial se ha completado correctamente. Incluso cuando el usuario utiliza otros modos (AR o Street View), el sistema siempre retorna a NoAR como estado estable.

Desde el punto de vista operativo, esta es la escena que **más va a usar y revisar el DDTI**, ya que concentra la visualización estándar del modelo y la mayor parte de la interacción.

Función general de la escena

En la escena NoAR, el modelo 3D del campus ya se encuentra cargado en memoria y es instanciado para su visualización directa. El usuario puede explorar el campus sin necesidad de hardware compatible con realidad aumentada, lo que garantiza que la aplicación sea funcional en cualquier dispositivo soportado.

Las responsabilidades principales de esta escena son:

- Mostrar el modelo 3D completo del campus.
- Permitir la navegación mediante cámara.
- Permitir la selección de edificios.
- Servir como punto de acceso a AR y Street View.
- Mantener la interfaz principal de la aplicación.

NoAR **no ejecuta procesos de carga pesada**, solo reutiliza el contenido que ya fue preparado en la escena Loading.

10.2.1 Script: ModelRestorer

En esta escena, ModelRestorer se encarga de instanciar el modelo 3D desde el AssetBundle que ya fue cargado previamente. El script verifica que el contenido esté disponible y, si aún no existe una instancia del modelo en la escena, la crea automáticamente.

Una vez instanciado, el modelo queda disponible para todos los sistemas de interacción y visualización. En NoAR, no se aplican ajustes especiales de posicionamiento relacionados con la cámara real, ya que se trata de un entorno 3D controlado.

```

void Start()
{
    AssetBundleLoader.child = child;
    if (AssetBundleLoader.Instance != null && instantiateOnStart)
    {
        InstantiateModel();
    }

    if (SceneManager.GetActiveScene().name.Equals("AR"))
    {
        StartCoroutine(SetARModel());
    }
}

```

Si este script falla:

- El modelo 3D no aparece en la escena.
- La interfaz puede mostrarse, pero la aplicación queda sin contenido útil.
- El problema no está en la carga, sino en la instanciación o referencias internas.

10.2.2 Script: UIManager

UIManager controla toda la interfaz visible en la escena NoAR. Desde aquí se gestionan los menús, botones y ventanas informativas que permiten al usuario interactuar con la aplicación.

```

1      using System.Collections;
2      using System.Collections.Generic;
3      using UnityEngine;
4      using UnityEngine.ResourceManagement.AsyncOperations;
5      using UnityEngine.ResourceManagement.ResourceProviders;
6      using UnityEngine.SceneManagement;
7      using UnityEngine.UI;
8
9      [System.Serializable]
10     > public class MenuConfigs...
35
36     [System.Serializable]
37     > public class Menus...
68
69     > public class UIManager...
596

```

En esta escena, el UIManager:

- Muestra la interfaz principal.
- Permite cambiar entre orientaciones vertical y horizontal.
- Gestiona la selección de edificios.
- Habilita o deshabilita el acceso a AR según compatibilidad.

- Permite el acceso al modo Street View.

Si este script presenta fallos, los síntomas típicos son:

- Botones que no responden.
- Menús que no se abren o no se cierran.
- Imposibilidad de cambiar de modo o escena.

10.2.3 Script: CameraController

El CameraController permite al usuario navegar por el modelo 3D utilizando gestos táctiles. En NoAR, la cámara es el principal medio de exploración del campus.

```
void Update()
{
    InputInteractions();
}

1 referencia
private void InputInteractions()...

1 referencia
private void MoveCamera()...

1 referencia
private void RotateCameraAround()...

0 referencias
private void InclineCamera()...

2 referencias
private float GetRadians(Vector2 delta)...

1 referencia
private void ScaleGameObject()...
```

Este script controla:

- Desplazamiento de la cámara.
- Rotación.
- Zoom, dentro de límites predefinidos.

Si el script no funciona correctamente:

- El modelo se muestra, pero no puede explorarse.
- El usuario percibe la aplicación como “congelada” aunque no lo esté.

10.2.4 Script: InteractionManager

InteractionManager permite la interacción directa con el modelo 3D. Detecta los toques del usuario y determina qué edificio ha sido seleccionado.

```
void Update()
{
    InputInteractions();
}

1 referencia
private void InputInteractions()...

1 referencia
private void TouchLocation(Touch touch)...

2 referencias
private void ToggleTitleBackground(GameObject location)...

0 referencias
private void SetTitleBackground(GameObject location, bool active)...

1 referencia
private IEnumerator SetSelectedTitle(LocationTitle title)...

1 referencia
IEnumerator InteractionDelay(Touch touch)...

1 referencia
IEnumerator MoveToVector3(Vector3 newPosition)...
```

En esta escena, este script:

- Detecta taps y doble taps.
- Realiza raycasts desde la cámara.
- Identifica edificios mediante su configuración (colliders y tags).
- Informa al UIManager qué edificio está seleccionado.

Cuando este script falla:

- No se pueden seleccionar edificios.
- No se actualiza la información mostrada en la interfaz.
- La navegación visual funciona, pero la interacción lógica no.

10.3 Escena AR

La escena **AR** es el modo de realidad aumentada de la aplicación. No es el modo base: solo se utiliza cuando el usuario lo solicita y cuando el dispositivo confirma compatibilidad. Si el dispositivo no soporta AR, la aplicación debe permanecer en NoAR sin intentar entrar a esta escena.

En términos de mantenimiento, esta escena depende de tres cosas para funcionar bien: que el contenido ya esté listo (AssetBundle cargado), que el dispositivo tenga soporte AR, y que la UI gestione correctamente el acceso y retorno.

Función general de la escena

En AR, el modelo 3D del campus se visualiza sobre el entorno real capturado por la cámara del dispositivo. El usuario puede manipular el modelo con gestos para ajustarlo a su preferencia (rotación y escala). La escena está pensada para ofrecer una experiencia “inmersiva”, pero su funcionamiento no debe afectar el modo NoAR.

Esta escena no vuelve a cargar ni descifrar contenido. Reutiliza el AssetBundle ya disponible en memoria.

10.3.1 Script: ModelRestorer

En AR, ModelRestorer cumple el mismo objetivo que en NoAR: instanciar el modelo 3D desde el AssetBundle. La diferencia es que aquí ajusta el modelo para que se ubique de forma adecuada en relación con la cámara, ya que la referencia visual ahora es el entorno real.

```
private IEnumerator SetARModel()
{
    yield return new WaitForSeconds(0.5f);

    GameObject asset = AssetBundleLoader.Instance.modelInstance.gameObject;
    asset.transform.position = Camera.main.transform.TransformPoint(pos);
    Vector3 targetPosition = Camera.main.transform.position;
    targetPosition.y = asset.transform.position.y;
    asset.transform.LookAt(targetPosition);
    ARModelSet = true;
}
```

Si este script falla en AR:

- La escena puede abrir, pero no aparece el modelo.
- El usuario ve la cámara, la UI, pero sin contenido 3D.
- Es un fallo de instanciación/posición, no del cifrado (si NoAR sí funciona).

10.3.2 Script: UIManager

En la escena AR, UIManager mantiene el control de la interfaz. Aquí es especialmente importante porque:

- Permite volver a NoAR.
- Controla opciones específicas del modo.
- Puede activar funciones como compartir captura.

Si falla:

- El usuario puede quedar atrapado en la escena AR sin navegación clara.
- Botones de salida o menús dejan de funcionar.

10.3.3 Script: ARSupport

ARSupport es el componente que determina si el dispositivo soporta realidad aumentada. Este script se ejecuta al iniciar y guarda el resultado para que el sistema decida si permite el modo AR o si debe bloquearlo.

En la práctica, este script es el “filtro” que evita que un dispositivo incompatible intente ejecutar AR.

```
private IEnumerator Start()
{
    IsChecked = false;
    IsSupported = false;

    if (ARSession.state == ARSessionState.None)
        yield return ARSession.CheckAvailability();

    IsSupported =
        ARSession.state == ARSessionState.Ready ||
        ARSession.state == ARSessionState.SessionInitializing ||
        ARSession.state == ARSessionState.SessionTracking;

    IsChecked = true;
}
```

Si falla:

- Puede habilitar AR en dispositivos no compatibles (termina en pantalla vacía, errores o comportamiento inestable).
- Puede bloquear AR en dispositivos compatibles (el usuario nunca puede usar el modo aunque su teléfono sí lo soporte).

10.3.4 Script: LocationController

En la escena AR, LocationController se encarga de la manipulación del modelo ya instanciado. Es el responsable de que el usuario pueda:

- Rotar el modelo con un dedo.

- Escalarlo con dos dedos.
- Resetear transformaciones cuando se requiera.

Este script también tiene funcionalidad opcional para generar carteles de edificios si está activada, pero en AR su rol principal es la manipulación estable del modelo usando un pivot.

Si falla:

- El modelo aparece, pero no se puede rotar o escalar.
- La experiencia se siente “rota” aunque el contenido esté cargado.

10.3.5 Script: ShareImage

ShareImage permite capturar una imagen de la pantalla y compartirla mediante el sistema operativo del dispositivo. En la implementación actual, esta funcionalidad está desarrollada para Android, y en iOS no ejecuta la lógica de compartir.

Si falla:

- No afecta la visualización ni la interacción del modelo.
- Solo se pierde la capacidad de compartir la captura.

10.4 Escena SV (Street View)

La escena **SV (Street View)** corresponde al modo de navegación mediante imágenes 360°. Este modo es complementario al modelo 3D y permite recorrer espacios específicos del campus como si se tratara de un recorrido virtual punto a punto. No es un modo base ni obligatorio; siempre se accede a él desde NoAR o AR y, al cerrarse, se retorna al modo principal.

Desde el punto de vista operativo, SV es un modo **independiente del modelo 3D**, pero dependiente del estado general de la aplicación y de la selección previa realizada por el usuario.

Función general de la escena

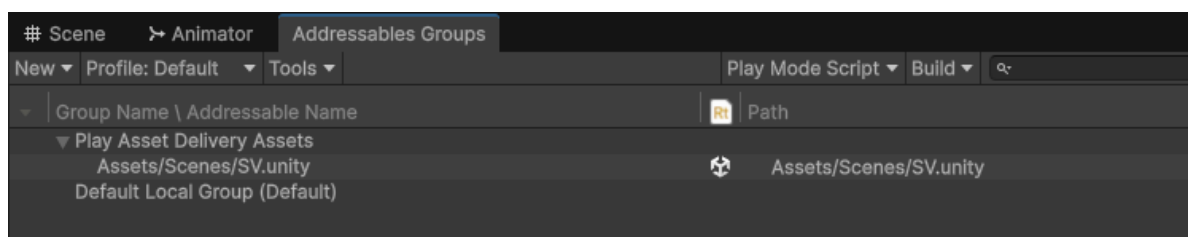
En esta escena no se renderiza el modelo 3D del campus. En su lugar, se muestra una imagen 360° aplicada sobre una geometría interna (esfera invertida), permitiendo al

usuario mirar en todas las direcciones. La navegación entre ubicaciones se realiza mediante nodos interactivos que representan puntos de transición entre stands.

La escena SV reutiliza información definida previamente (por ejemplo, el edificio seleccionado antes de entrar), pero no requiere volver a cargar ni descifrar contenido protegido.

10.4.1 Script: SVLoader

SVLoader controla la entrada y salida del modo Street View. Su función es cargar la escena SV cuando el usuario lo solicita y cerrarla cuando se decide regresar al modo principal.



El comportamiento varía según la plataforma:

- En Android, la escena SV puede cargarse y descargarse dinámicamente mediante Addressables o PAD.
- En iOS, la escena se gestiona directamente como una escena incluida en el build.

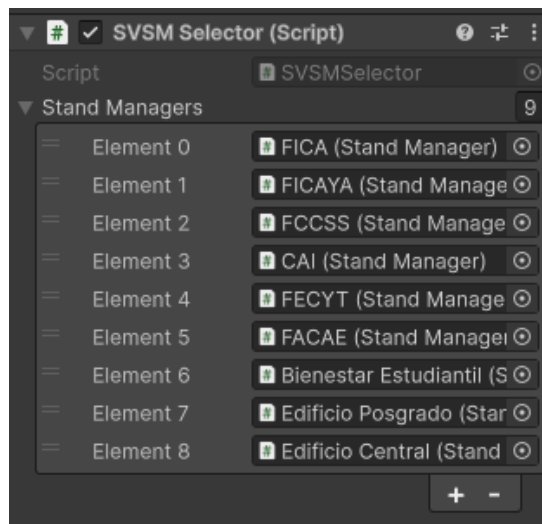
Si este script falla:

- No se puede entrar al modo SV.
- El usuario puede quedar atrapado en SV o no poder salir correctamente.
- El flujo de navegación se rompe.

10.4.2 Script: SVSMSelector

Este script se ejecuta al iniciar la escena SV y se encarga de activar únicamente el conjunto de datos (StandManager) correspondiente a la ubicación que el usuario seleccionó antes de entrar al modo Street View.

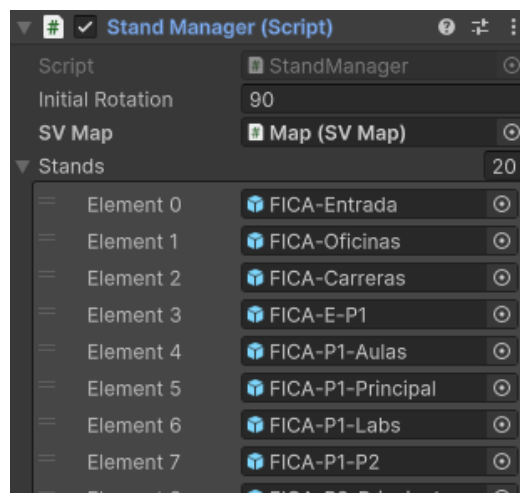
La selección se basa en el nombre de la ubicación, por lo que es importante que exista consistencia entre los nombres de los edificios y los managers configurados.



Si falla:

- La escena SV se carga, pero aparece vacía o sin puntos de navegación.
- En consola suele aparecer un mensaje indicando que no se encontró la ubicación.

10.4.3 Script: StandManager



StandManager es el controlador principal del modo Street View. Gestiona:

- Qué stand está activo.
- La detección de nodos tocados por el usuario.
- El movimiento de la cámara entre stands.
- La actualización de la imagen 360° mostrada.

Este script define el flujo interno del recorrido virtual. Si no funciona correctamente, el modo SV pierde completamente su utilidad.

10.4.4 Script: SMap

SMap es el encargado de aplicar la imagen 360° correspondiente al stand activo sobre la geometría de visualización. Cambia la textura y los parámetros necesarios para que la imagen se muestre correctamente.

```
public void SetMap(Texture2D texture, Vector2 offset)
{
    meshRenderer.material.SetTexture("_MainTex", texture);
    meshRenderer.material.SetTextureOffset("_MainTex", offset);
}
```

Si falla:

- La escena se muestra, pero la imagen no cambia.
- El usuario puede ver una imagen incorrecta o una esfera sin textura.

10.4.5 Script: Stand

El script Stand define los datos de cada punto del recorrido. Contiene la referencia a la imagen 360° y los parámetros necesarios para su correcta visualización.

No ejecuta lógica compleja en tiempo de ejecución; su función principal es almacenar información que será utilizada por StandManager.

```
private void SetStand()
{
    gameObject.GetComponentInParent<StandManager>().DisableStands();
    gameObject.SetActive(true);
    StandManager standManager = gameObject.GetComponentInParent<StandManager>();
    standManager.sVMap.SetMapOutOfRuntime(texture, offset);
}
```

10.4.6 Script: Node

Node representa un punto interactivo dentro del recorrido Street View. Cada nodo enlaza hacia un stand destino. Al tocar un nodo, el sistema sabe a qué ubicación debe moverse.

```
public class Node : MonoBehaviour
{
    public Stand stand;
}
```

Si falla:

- El usuario puede ver nodos, pero no ocurre ningún cambio al tocarlos.

- La navegación entre stands se rompe.

10.4.7 Script: Sign

Sign controla los carteles informativos asociados a los stands. Se encarga de mostrar el nombre del punto y de orientar el cartel hacia la cámara para que sea legible.

```
public class Sign : MonoBehaviour
{
    public Text label;

    [ContextMenu("InstallSign")]
    1 referencia
    public void InstallSign()...
```

Si falla:

- La navegación sigue funcionando.
- Se pierde información visual y claridad para el usuario.

11. Interacción del usuario

Este apartado describe cómo el usuario interactúa con la aplicación una vez que el contenido ya está cargado y visible. La interacción se basa exclusivamente en **gestos táctiles** y en el uso de la **interfaz gráfica**, sin requerir periféricos adicionales ni configuraciones especiales.

11.1 Gestos en NoAR (cámara)

En el modo NoAR, la interacción principal del usuario se realiza a través del control de la cámara que observa el modelo 3D del campus. El usuario puede desplazarse visualmente por el entorno utilizando gestos táctiles estándar.

Con un solo dedo, el usuario puede mover la cámara en el plano, permitiendo recorrer el campus desde diferentes ángulos. Con dos dedos, puede rotar la vista y realizar zoom dentro de límites previamente definidos. Estos límites evitan que la cámara se acerque demasiado o se aleje en exceso, manteniendo una experiencia controlada.

11.2 Gestos en AR (manipulación del modelo)

En el modo AR, la interacción cambia de enfoque. En lugar de mover una cámara virtual, el usuario manipula directamente el modelo 3D sobre el entorno real capturado por la cámara del dispositivo.

Con un dedo, el usuario puede rotar el modelo. Con dos dedos, puede anclarlo a la cámara para moverlo y escalarlo para ajustarlo al espacio físico disponible. Estas interacciones no afectan la posición real del dispositivo, solo transforman el modelo virtual.

11.3 Selección de edificios (tap / doble tap)

La selección de edificios se realiza principalmente en el modo NoAR. El usuario puede tocar directamente sobre un edificio del modelo 3D para seleccionarlo. Al hacerlo, la aplicación identifica el edificio y actualiza la interfaz mostrando su nombre y la información asociada.

El doble tap tiene una función adicional: centra la cámara en el edificio seleccionado, facilitando su visualización. Este comportamiento es intencional y forma parte de la experiencia de exploración.

11.4 Interacción en Street View (SV)

En el modo Street View, la interacción del usuario no se basa en la manipulación de un modelo 3D, sino en la navegación dentro de un entorno de imágenes 360°. El comportamiento esperado es distinto al de NoAR y AR, y por tanto los gestos tienen una interpretación específica.

El desplazamiento visual se realiza arrastrando un dedo sobre la pantalla, lo que permite al usuario mirar alrededor del entorno 360° en cualquier dirección. Este gesto no provoca movimiento entre ubicaciones, únicamente cambia el ángulo de visión dentro del mismo stand.

La navegación entre ubicaciones se realiza tocando los nodos visibles dentro de la escena. Cada nodo representa un punto de transición hacia otro stand. Al tocar un nodo, la aplicación debe cambiar la imagen 360° activa y reposicionar la cámara en el nuevo punto.

12. Realidad Aumentada

Este apartado describe **cómo funciona el modo AR, cuándo se habilita, cuándo no, y qué comportamiento es el correcto** desde el punto de vista operativo. No entra en detalles de implementación interna; se centra en criterios de uso y diagnóstico.

12.1 Detección de compatibilidad (ARSupport)

El acceso al modo de Realidad Aumentada **no es automático**. Antes de permitir la entrada a la escena AR, la aplicación verifica si el dispositivo es compatible.

Esta verificación se realiza al iniciar la aplicación y su resultado se conserva para el resto de la ejecución. El sistema evalúa si el dispositivo soporta AR a nivel de hardware y sistema operativo.

El comportamiento esperado es el siguiente:

- Si el dispositivo **es compatible**, el botón o acceso a AR se habilita.
- Si el dispositivo **no es compatible**, el acceso a AR permanece deshabilitado y la aplicación continúa funcionando únicamente en modo NoAR.

En ningún caso la aplicación debe forzar la entrada a AR en un dispositivo no compatible.

12.2 Comportamiento en dispositivos no compatibles

Cuando un dispositivo no soporta Realidad Aumentada, este **no se considera un error** ni un fallo del sistema. Es un escenario previsto.

En estos casos:

- El usuario debe permanecer en el modo NoAR.
- La aplicación debe seguir siendo completamente funcional.
- No deben mostrarse pantallas vacías, errores ni intentos de inicializar la cámara AR.

Si la aplicación intenta abrir AR y falla visualmente, o se muestra una pantalla negra o sin modelo en un dispositivo no compatible entonces el problema está en la detección de compatibilidad y no en la carga del contenido ni en el modelo 3D.

13. Street View 360

Este apartado describe el funcionamiento del modo **Street View (SV)** desde un enfoque operativo. Street View es un modo adicional de visualización que permite recorrer espacios específicos del campus mediante imágenes 360°. No es un modo principal ni obligatorio para el uso de la aplicación.

13.1 Estructura de stands, nodos y carteles

El modo Street View se basa en una estructura jerárquica simple. Cada ubicación recorrible se denomina *stand*. Un stand contiene una imagen 360° que representa un punto específico del campus.

Dentro de cada stand existen nodos visibles para el usuario. Estos nodos funcionan como puntos de transición y permiten moverse hacia otros stands relacionados. La relación entre stands define el recorrido posible dentro del modo Street View.

Los carteles o *signs* se utilizan únicamente como apoyo visual. Muestran el nombre de la ubicación o información contextual y siempre están orientados hacia la cámara para mantener la legibilidad. Los carteles no influyen en la navegación ni en la lógica del sistema.

13.2 Flujo de selección de nodo y cambio de stand

El flujo de navegación en Street View es directo. El usuario observa el entorno arrastrando un dedo para cambiar el ángulo de visión. Cuando desea desplazarse a otra ubicación, toca un nodo visible en la escena.

Al seleccionar un nodo, el sistema identifica el stand de destino asociado y realiza el cambio de imagen 360°. Este cambio ocurre sin recargar escenas ni reiniciar el sistema; únicamente se actualiza el contenido visual correspondiente.

Si el usuario puede mirar alrededor pero no puede cambiar de ubicación, el problema suele estar en la detección de nodos o en la relación entre stands. Si el cambio ocurre pero la imagen es incorrecta, el problema está en la configuración del stand o en la imagen asociada.

13.3 Diferencias entre Android e iOS

El modo Street View presenta diferencias internas según la plataforma, aunque el comportamiento visible para el usuario es el mismo.

En Android, el sistema puede utilizar mecanismos de carga dinámica de contenido (como Addressables o Play Asset Delivery) para gestionar recursos del modo Street View. Esto permite reducir el tamaño inicial de la aplicación y cargar recursos solo cuando se necesitan.

En iOS, el contenido de Street View se gestiona directamente como parte del proyecto y se carga mediante el sistema estándar de escenas. No se utiliza carga dinámica externa debido a las restricciones de la plataforma.

Estas diferencias no afectan el uso ni el mantenimiento cotidiano del modo Street View, pero deben tenerse en cuenta al momento de compilar o modificar el proyecto para una plataforma específica.

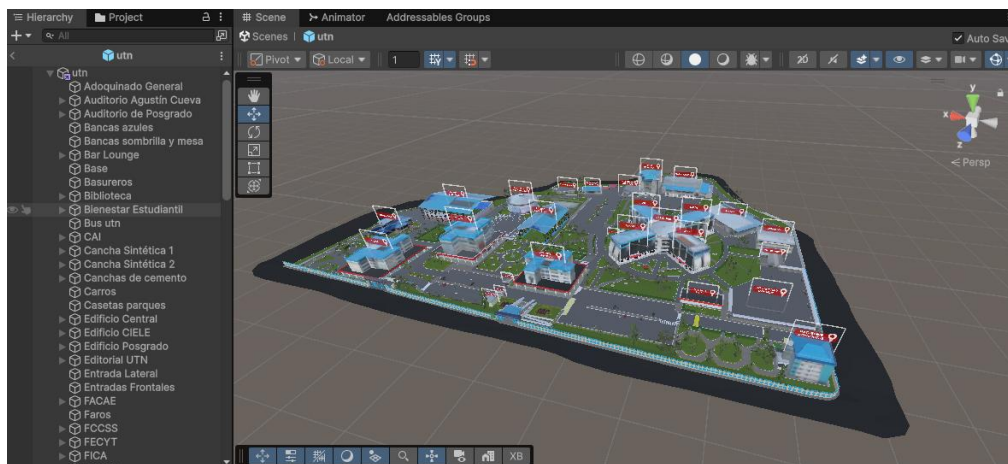
14. Procedimientos de mantenimiento

14.1 Actualizar el modelo 3D

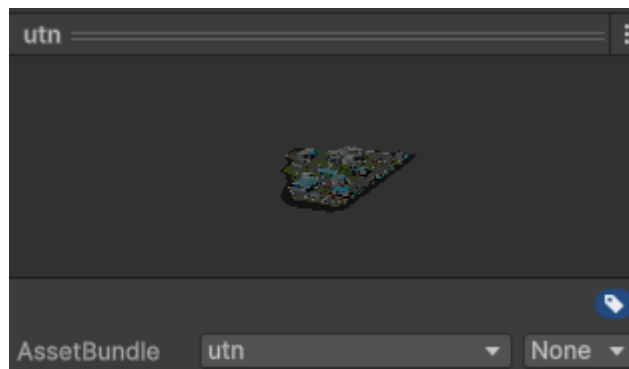
Este procedimiento se utiliza cuando el campus cambia (nuevos edificios, ajustes de geometría, correcciones visuales).

Acciones permitidas:

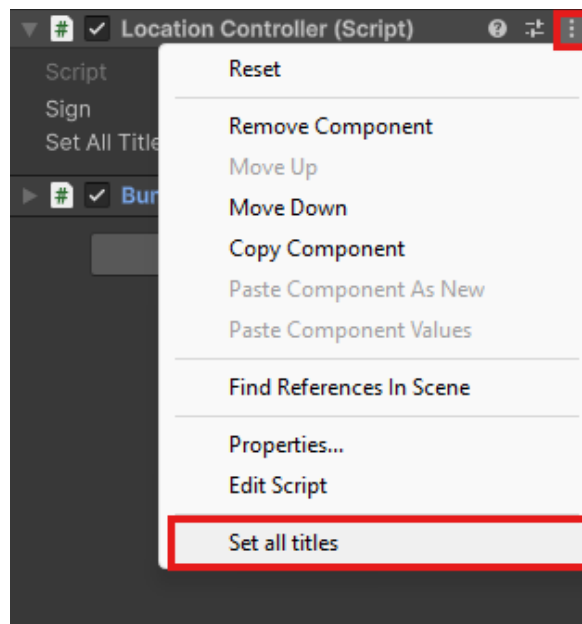
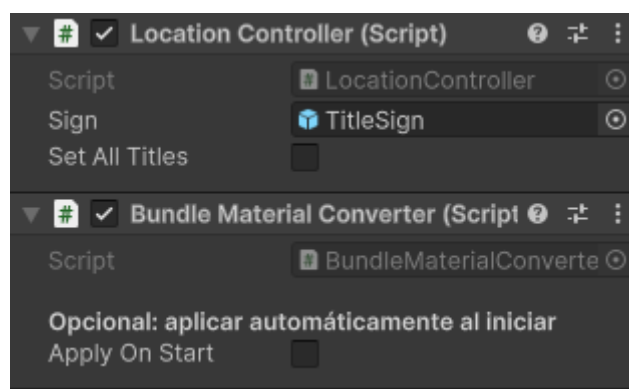
- Reemplazar o actualizar el modelo 3D base.
- Ajustar jerarquía, colliders y scripts asociados al modelo.



- Asignar Prefab como AssetBundle (parte inferior del inspector).



- Asignar scripts correspondientes e inicializar funciones.



- Verificar que el modelo funcione correctamente dentro del Editor antes de continuar.

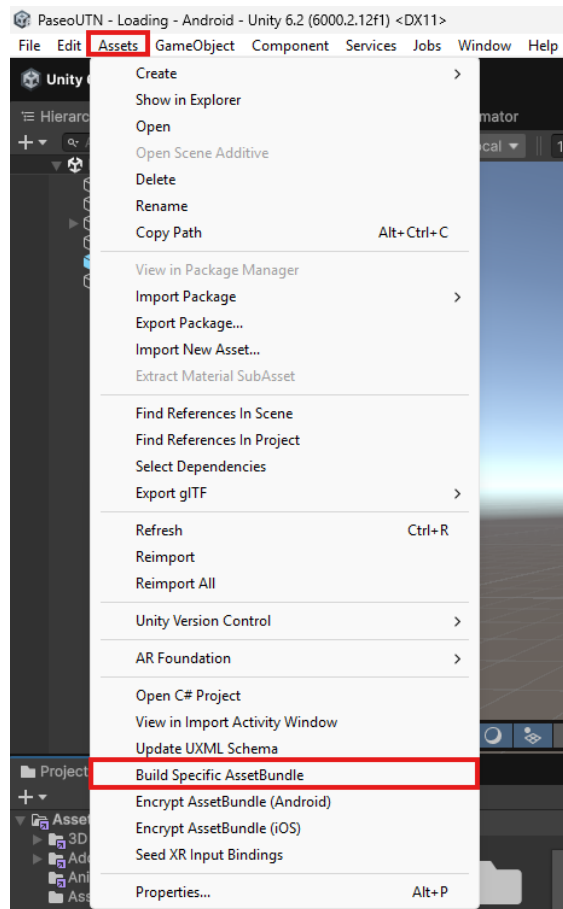
Acciones no permitidas:

- Incluir el modelo directamente en escenas finales.
- Omitir el proceso de AssetBundle y cifrado.

Resultado esperado:

- Un **Prefab único** del campus listo para empaquetar.

14.2 Regenerar el AssetBundle



Este paso es obligatorio cada vez que el modelo 3D cambia.

Procedimiento:

- Abrir el proyecto en Unity.
- Ejecutar el script de generación de AssetBundles (CreateAssetBundles).
- Verificar que el AssetBundle se genere sin errores.

Consideraciones:

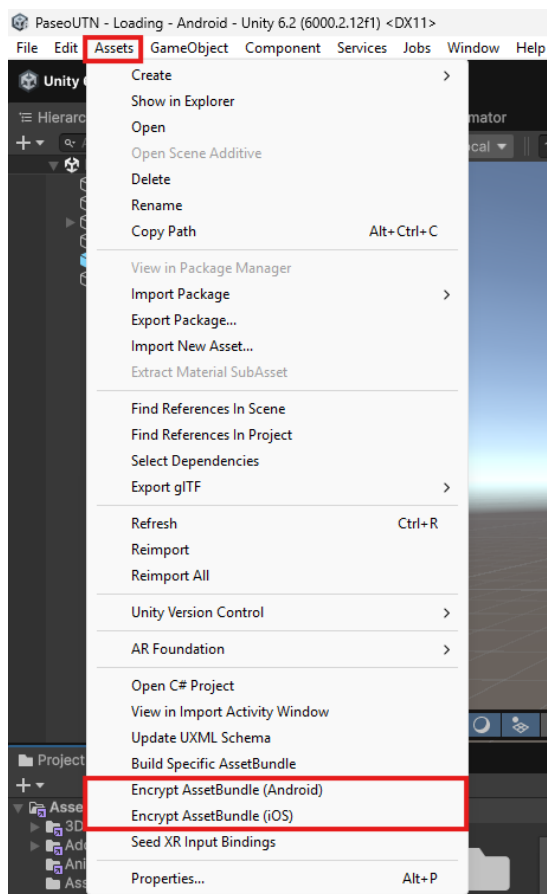
- El AssetBundle generado aún **no es seguro**.

- No debe copiarse directamente a la aplicación final.

Resultado esperado:

- AssetBundle válido por plataforma.

14.3 Re-cifrar y reemplazar el archivo .enc



Este es el paso más crítico del mantenimiento.

Procedimiento:

- Ejecutar el script de cifrado AES (Encryptor AES) sobre el AssetBundle generado.
- Verificar que el proceso finalice correctamente.
- Copiar el archivo .enc resultante a la carpeta StreamingAssets.
- Reemplazar el archivo .enc anterior.

Reglas clave:

- La clave de cifrado **no debe cambiarse**.
- El nombre y la ruta del archivo .enc deben mantenerse.
- Nunca mezclar archivos de distintas versiones.

Resultado esperado:

- Un único archivo .enc válido y actualizado.

14.4 Probar antes de liberar

Antes de generar una versión final de la aplicación, se deben realizar pruebas mínimas obligatorias.

Pruebas recomendadas:

- La aplicación supera la escena Loading.
- El modelo aparece correctamente en NoAR.
- La cámara y selección funcionan.
- AR se habilita solo en dispositivos compatibles.
- Street View abre y permite navegación básica.

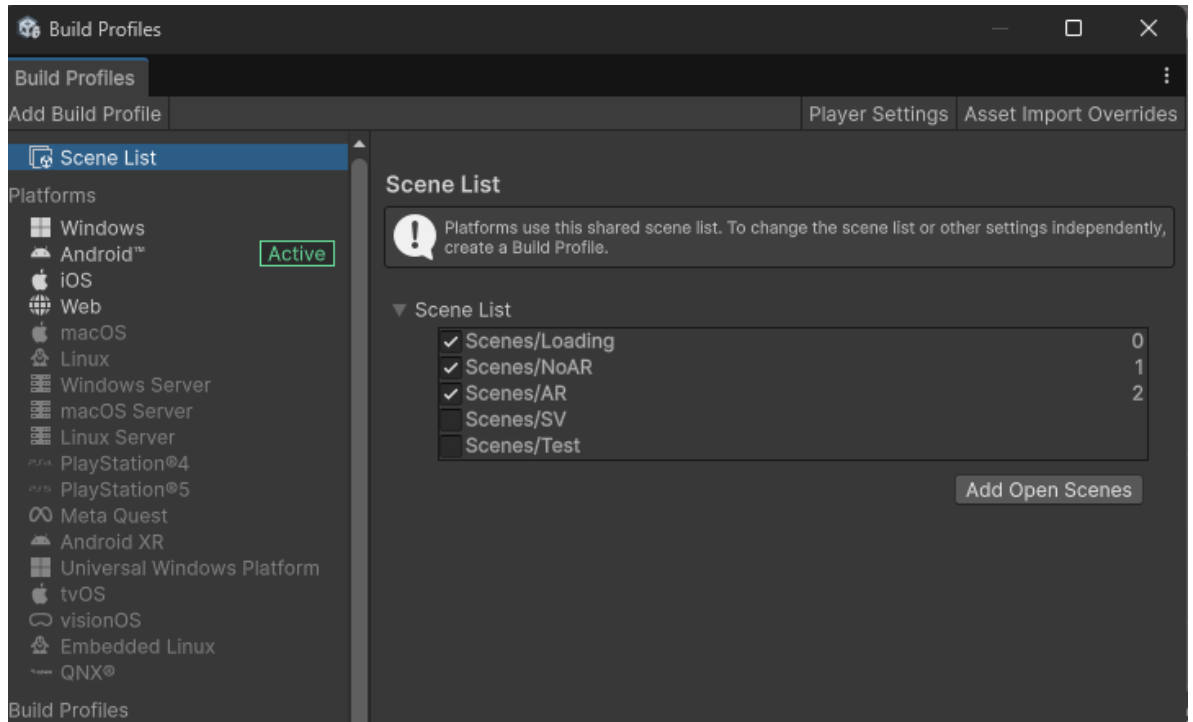
15. Compilación y despliegue

Este apartado describe cómo generar versiones instalables de la aplicación **Paseo UTN** para Android e iOS. Antes de compilar, es obligatorio verificar que el proyecto esté correctamente configurado y que el contenido protegido esté listo.

Para la publicación de iOS se recomienda seguir la siguiente guía: https://www.youtube.com/watch?v=8l2GT3_JiUU

Para la publicación de Android se recomienda seguir la siguiente guía: https://www.youtube.com/watch?v=UXl_C3ZnRLc

15.1 Escena inicial obligatoria (Loading)



Antes de realizar cualquier build, se debe verificar que la escena **Loading** esté configurada como la **primera escena del build**.

Esta escena **siempre debe ir en la posición 0** de la lista de escenas, ya que es la responsable de ejecutar el pipeline de carga y descifrado del contenido. Si otra escena se coloca antes de Loading, la aplicación no podrá cargar el modelo 3D y quedará en un estado inconsistente.

Este punto aplica tanto para Android como para iOS y **no debe omitirse** bajo ninguna circunstancia.

15.2 Configuraciones técnicas por plataforma

Las configuraciones específicas de cada sistema operativo se realizan desde:

Edit → Project Settings → Android / iOS

Desde este apartado se pueden modificar, según la plataforma:

- Bundle Identifier
- Versión de la aplicación
- Ícono de la app
- Versión mínima del sistema operativo
- APIs objetivo (Android)

- Orientación de pantalla
- Opciones varias

Estos cambios no afectan la lógica interna de la aplicación, pero son obligatorios para cumplir con los requisitos de Google Play y App Store.

15.3 Proceso de build (ambas plataformas)

El proceso de compilación se inicia desde:

File → Build Profiles

Desde aquí se selecciona la plataforma destino y se ejecuta el build.

Consideración crítica sobre escenas

- **Android**
La escena **SV (Street View)** **NO debe estar marcada** en la lista de escenas del build, ya que se carga mediante Addressables / Play Asset Delivery.
- **iOS**
La escena **SV SÍ debe estar marcada** en la lista de escenas del build, ya que en iOS no se utiliza Addressables y la escena debe incluirse directamente.

Un error en esta configuración provoca fallos directos en el modo Street View.

15.4 Firma y despliegue en Android

Para Android, la aplicación debe firmarse correctamente para ser aceptada por Google Play.

La firma se configura en:

Edit → Project Settings → Android → Publishing Settings

Es obligatorio:

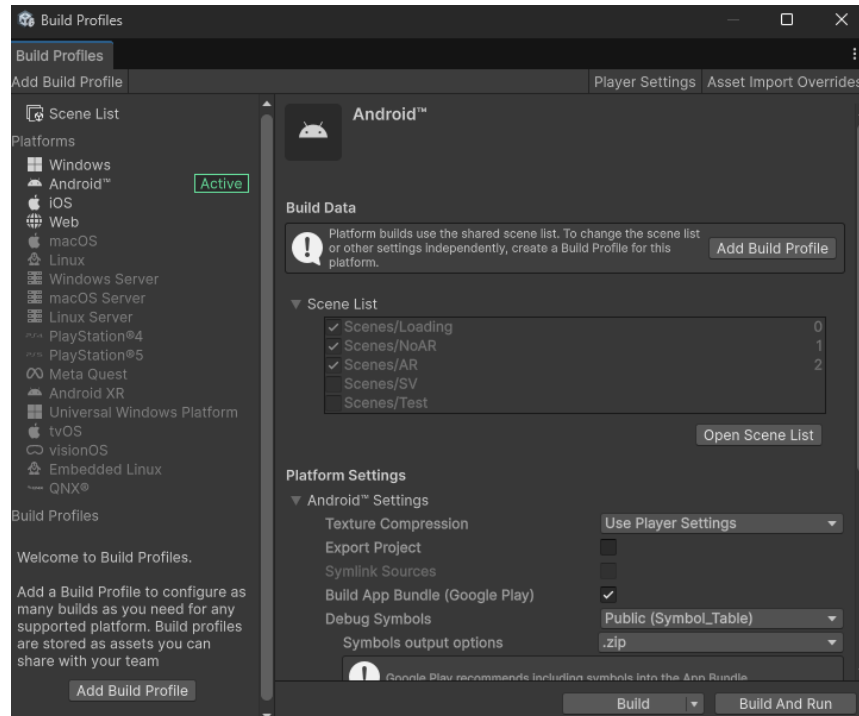
- Ingresar la contraseña del proyecto: *yT9\$kJ3p!bR@8vM2wQxZ*
- Usar **la misma contraseña** para:
 - Project Keystore Password
 - Project Key Password

Si las contraseñas no coinciden o no están configuradas, Google Play rechazará la subida indicando que la firma es incorrecta.

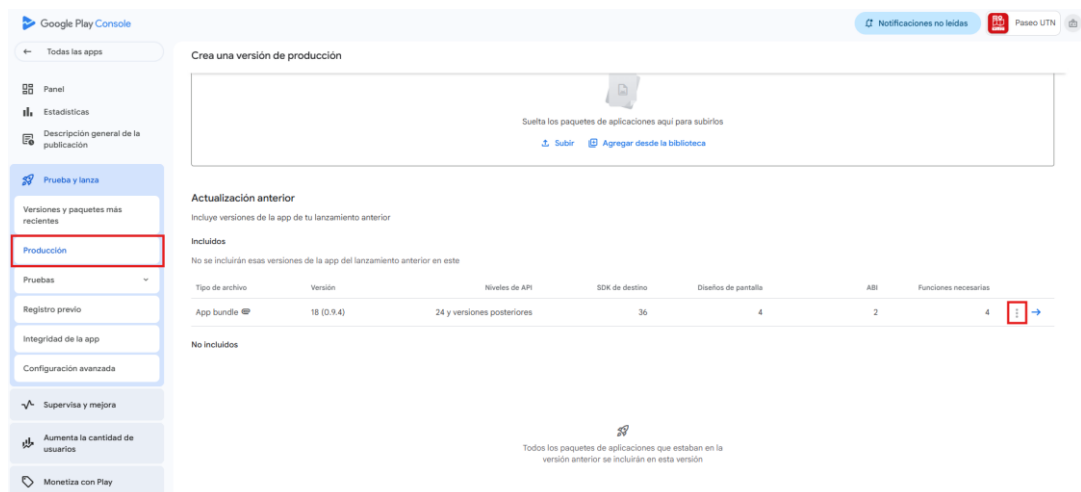
Generación del archivo .aab

Para obtener el archivo requerido por Google Play:

- En **Build Profiles** de Android:
 - Activar Build App Bundle (Google Play)
 - Configurar Debug Symbols como Public (Symbol_Table)



El archivo .aab generado debe subirse al crear una nueva versión en Google Play Console.



Subir archivo de asignación de ReTrace (.txt o .map)

Subir símbolos de depuración nativos (.zip)

Quitar app bundle

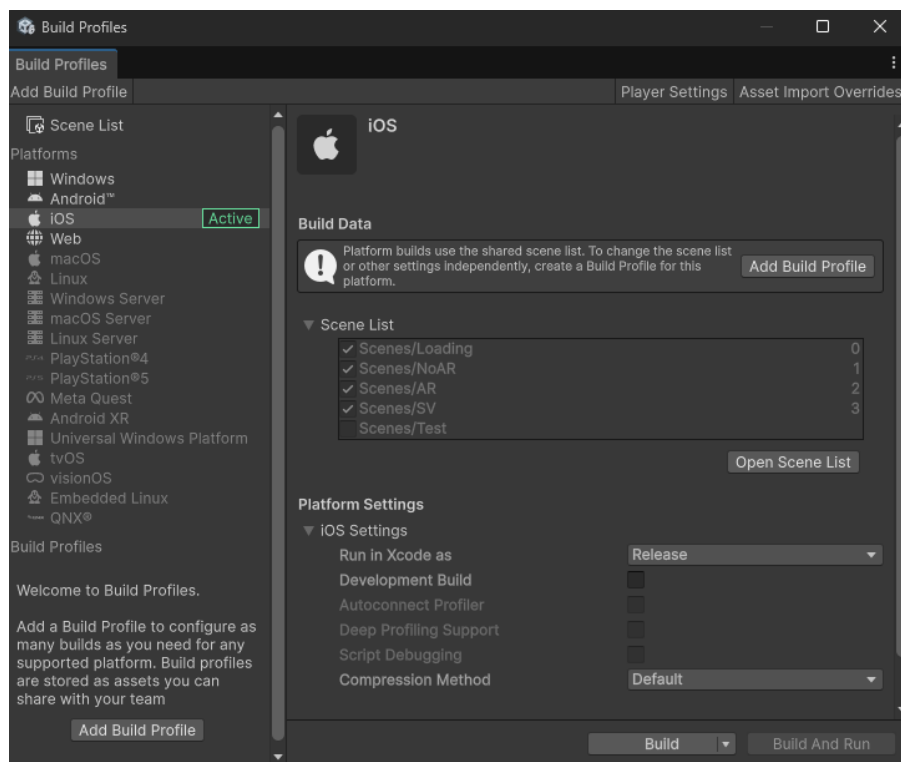


15.5 Firma y despliegue en iOS

La compilación para iOS se realiza en dos etapas.

Etapas Unity

- Generar el proyecto iOS desde Unity.
- Verificar que el archivo .enc esté presente en StreamingAssets.
- Confirmar que Loading sea la escena inicial.



Etapas Xcode

- Abrir el proyecto generado en Xcode.
- Asignar un perfil de desarrollo válido.
- Configurar la firma con la cuenta Apple correspondiente.

- Ejecutar el proceso estándar de **Archive**.
- Enviar la build a **App Store Connect**.

16. Evaluación del rendimiento del desarrollo

Las pruebas de rendimiento del desarrollo de la aplicación **Paseo UTN** se realizaron utilizando una laptop con **32 GB de memoria RAM, procesador AMD Ryzen 7 serie 4000 y tarjeta gráfica NVIDIA GTX 1650 Ti**, como entorno principal de desarrollo en Unity. Para las pruebas de ejecución en dispositivo móvil se utilizó un **Google Pixel 9**, con el fin de evaluar el comportamiento del sistema en un entorno real de uso.

La evaluación se enfoca en métricas técnicas internas del proceso de desarrollo, tales como tiempos de compilación, carga de escenas, uso de recursos, escalabilidad del contenido y mantenibilidad del código, obtenidas durante las fases de construcción y prueba del proyecto

16.1 Métricas evaluadas

La evaluación del rendimiento del desarrollo se realizó a partir de métricas técnicas internas obtenidas durante el proceso de construcción y prueba del proyecto.

16.1.1 Tiempo de compilación

Mide el tiempo requerido para generar una compilación del proyecto en Unity.

Métricas registradas:

- Tiempo promedio de compilación: 7 min
- Tiempo de compilación incremental: 4 min

16.1.2 Tiempo de carga de escenas

Evalúa el tiempo necesario para cargar las escenas principales del sistema.

Métricas registradas:

- Carga inicial de la aplicación: 4 s
- Cambio de modo NoAR a AR: 1.5 s
- Carga de recorridos interiores: 3.5 s

16.1.3 Uso de recursos

Analiza el consumo de recursos del sistema durante la ejecución de la aplicación en pruebas de desarrollo.

Métricas registradas:

- Uso promedio de memoria RAM: 700 MB
- Uso promedio de CPU: 30 %
- Variación con AR activado: 35 %

16.1.4 Tamaño del proyecto y compilaciones

Evalúa el impacto del tamaño del proyecto y de los archivos generados.

Métricas registradas:

- Tamaño total del proyecto Unity: 180.5 GB
- Tamaño de la compilación final: 298 MB
- Reducción mediante AssetBundles: 50 %

16.1.5 Escalabilidad del contenido

Mide la capacidad del sistema para integrar nuevo contenido sin afectar significativamente el rendimiento.

Métricas registradas:

- Número de edificios probados: 7
- Tiempo de integración por edificio: 10 h
- Impacto en tiempo de carga: 1 %

16.1.6 Mantenibilidad del código

Evalúa de forma descriptiva la estructura y organización del código fuente.

Métricas registradas:

- Número de scripts principales: 17
- Organización modular: media

Consideraciones a futuro

Como trabajo futuro del proyecto **Paseo UTN**, se plantean mejoras orientadas a aumentar la escalabilidad del sistema, enriquecer la experiencia del usuario y facilitar el mantenimiento del contenido sin comprometer la seguridad de la información.

Conexión con una base de datos pública sin datos sensibles

Se propone integrar una base de datos pública que contenga información general del campus (nombres de edificios, descripciones, horarios de atención, ubicaciones de oficinas, puntos de interés y avisos). Esta base debe excluir datos sensibles y personales, permitiendo que el contenido pueda actualizarse sin necesidad de recompilar la aplicación. Además, se recomienda incluir mecanismos de versionamiento y validación de datos para evitar inconsistencias entre la información mostrada en la aplicación y la almacenada en la base.

Implementación del módulo de búsqueda desarrollado

El proyecto cuenta con un módulo de búsqueda ya desarrollado, pero no implementado en la versión actual. Como mejora futura, se plantea su integración para permitir que el usuario localice rápidamente edificios, áreas internas, oficinas y puntos importantes. Este módulo podría complementarse con filtros por categoría (facultades, laboratorios, servicios, áreas deportivas) y sugerencias automáticas al escribir, mejorando la navegación sin depender únicamente del recorrido manual.

Activación de la funcionalidad de alternar calidad gráfica

Se plantea habilitar la funcionalidad ya desarrollada para alternar la calidad gráfica. Esto permitiría adaptar la aplicación a distintos perfiles de dispositivos, ofreciendo un modo de alto rendimiento (bajo consumo) y un modo de alta calidad (mayor detalle visual). Para que sea útil, se recomienda asociar esta opción a cambios concretos como resolución de texturas, distancia de dibujado, sombras, nivel de detalle (LOD) y desactivación de elementos secundarios en dispositivos de gama baja.

Desarrollo de módulos para otros campus

Como ampliación del alcance, se propone desarrollar módulos adicionales para incluir los demás campus de la universidad. La arquitectura modular del proyecto facilita este crecimiento, ya que cada campus puede incorporarse como contenido independiente (escenas, modelos o paquetes descargables). Esto permitiría escalar el sistema sin afectar el funcionamiento base de la aplicación y mantener una distribución eficiente del contenido según la necesidad del usuario.

Vinculación de lugares internos con la base de datos para guía y navegación

Se recomienda vincular los lugares internos de los edificios (aulas, laboratorios, departamentos, oficinas, servicios) con la base de datos para implementar funciones de guía y navegación. Esto permitiría que el usuario seleccione un destino y reciba una orientación dentro del edificio, ya sea mediante rutas sugeridas, puntos de referencia o navegación por “saltos” similares a Street View. Para esto, es clave definir una estructura de datos consistente (identificadores únicos por lugar, edificio y campus) y un sistema de relaciones que permita ubicar cada punto correctamente dentro del modelo.