



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Carrera de Computación

Desarrollo de realidad virtual inmersiva aplicada al entorno virtual de la carrera Computación e integración de respuestas inteligentes mediante la implementación de Inteligencia Artificial

Trabajo de Integración Curricular
previa a la obtención del título de
Ingeniero en Ciencias de la
Computación

AUTORES:

Macas Vélez José Andrés

Bryan Jeremy Encalada Mejía

DIRECTORES:

Ing. Andrés Roberto Navas Castellanos, Mg. Sc.

Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc.

Loja – Ecuador

2025

Certificación de director



unl

Universidad
Nacional
de Loja

Sistema de Información Académico
Administrativo y Financiero - SIAAF

CERTIFICADO DE CULMINACIÓN Y APROBACIÓN DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Nosotros, **ANDRES ROBERTO NAVAS CASTELLANOS, PABLO FERNANDO ORDOÑEZ ORDOÑEZ**, directores del Trabajo de Integración Curricular denominado **Desarrollo de realidad virtual inmersiva aplicada al entorno virtual de la carrera Computación e integración de respuestas inteligentes mediante la implementación de Inteligencia Artificial**, perteneciente al estudiante **JOSE ANDRES MACAS VELEZ**, con cédula de identidad N° **1105596918**.

Certifico:

Que luego de haber dirigido el **Trabajo de Integración Curricular**, habiendo realizado una revisión exhaustiva para prevenir y eliminar cualquier forma de plagio, garantizando la debida honestidad académica, se encuentra concluido, aprobado y está en condiciones para ser presentado ante las instancias correspondientes.

Es lo que puedo certificar en honor a la verdad, a fin de que, de así considerarlo pertinente, el/la señor/a docente de la asignatura de **Integración Curricular**, proceda al registro del mismo en el Sistema de Gestión Académico como parte de los requisitos de acreditación de la Unidad de Integración Curricular del mencionado estudiante.

Loja, 4 de Febrero de 2025



Firmado electrónicamente por:
ANDRES ROBERTO
NAVAS CASTELLANOS

F)

DIRECTOR DE TRABAJO DE INTEGRACIÓN CURRICULAR



Firmado electrónicamente por:
PABLO FERNANDO
ORDONEZ ORDOÑEZ

F)

DIRECTOR DE TRABAJO DE INTEGRACIÓN CURRICULAR



Certificado TIC/TT.: UNL-2025-000343

1/1
Educamos para **Transformar**

Autoría

Nosotros, **Bryan Jeremy Encalada Mejía**, y **José Andrés Macas Vélez**, declaramos ser autores del presente Trabajo de Integración Curricular y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja la publicación de nuestro Trabajo de Integración Curricular, en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Cédula de identidad: 1105596918

Fecha: 31-05-2025

Correo electrónico:

jose.a.macas.v@unl.edu.ec

Teléfono: 0994787866

Firma:

Cédula de identidad: 1105844995

Fecha: 31-05-2025

Correo electrónico:

bryan.encalada@unl.edu.ec

Teléfono: 0990504870

Carta de autorización por parte de los autores, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Integración Curricular

Nosotros, **Bryan Jeremy Encalada Mejía, y José Andrés Macas Vélez**, declaramos ser los autores del Trabajo de Integración Curricular denominado: **Desarrollo de realidad virtual inmersiva aplicada al entorno virtual de la carrera Computación e integración de respuestas inteligentes mediante la implementación de Inteligencia Artificial**, autorizamos al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Integración Curricular que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los 31 días del mes de marzo de dos mil veinticinco.

Firma:

Autor: José Andrés Macas Vélez

Cédula de identidad: 1105596918

Dirección: Manuel Quiroga y C-04-7.

Correo electrónico:

jose.a.macas.v@unl.edu.ec

Teléfono: 0994787866

Firma:

Autor: Bryan Jeremy Encalada Mejía

Cédula de identidad: 1105844995

Dirección: Celica y 18 de Noviembre.

Correo electrónico:

bryan.encalada@unl.edu.ec

Teléfono: 0990504870

DATOS COMPLEMENTARIOS:

Director del Trabajo de Integración Curricular: Ing. Andrés Roberto Navas Castellanos, Mg. Sc.

Director del Trabajo de Integración Curricular: Ing. Pablo Fernando Ordoñez Ordoñez, Mg. Sc.

Dedicatoria

Dedico este trabajo a mis padres, Diana Del Cisne Vélez Gaona y José Miguel Macas Jiménez, quienes fueron pilares fundamentales en cada etapa de este camino, su amor, apoyo incondicional y confianza en mí me impulsaron a seguir adelante, incluso en los momentos más desafiantes.

A mi hermana, Mabelyn Ximena Macas Vélez, quien me brindo su consejo, apoyo y compañía incondicional, brindándome momentos de alegría y tranquilidad en este proceso.

José Andrés Macas Vélez

A mis padres, Lilian Birmania Mejía Ordóñez y Ángel Geovanny Encalada Núñez, por ser mi pilar fundamental, por su amor incondicional, sacrificios y enseñanzas que me han guiado en cada paso de mi vida. Gracias por creer en mí y darme las herramientas para alcanzar mis metas.

A mis hermanas, Alisson Nicol Encalada Mejía y Emily Alejandra Encalada Mejía, por ser mi fuente de alegría, inspiración y compañía en este camino. Su apoyo y confianza han sido invaluableles en este proceso.

Este logro es también suyo.

Bryan Jeremy Encalada Mejía

Agradecimiento

A Adrián Nuñez, Joseph Ríos, Santiago Tene, Luis Jiménez, Omar Maldonado y a mi mejor amigo y compañero de trabajo, Jeremy Encalada, por su dedicación, esfuerzo y compañerismo a lo largo de este proyecto. Su compromiso y apoyo fueron fundamentales para alcanzar esta meta.

A toda la planta docente de la carrera de Computación de la Universidad Nacional de Loja, quienes con su enseñanza y guía me brindaron los conocimientos necesarios para desarrollarme en el ámbito profesional. En especial, agradezco a mi tutor, Ing. Andrés Roberto Navas Castellano, por su valiosa orientación, paciencia y apoyo durante la realización de este Trabajo de Integración Curricular.

Finalmente, a todas las personas que fueron parte de mi proceso de formación, tanto académica como personal, durante mi trayectoria universitaria. Su presencia, enseñanzas y motivación han sido esenciales en mi crecimiento y en la culminación de esta etapa tan importante en mi vida.

A todos ustedes, gracias.

Macas Vélez José Andrés

A lo largo de este camino, he contado con el apoyo de personas que han marcado mi vida y han hecho que este proceso sea más llevadero.

A mis amigos, aquellos que han estado presentes en cada momento, brindándome su compañía, aliento y risas en los momentos difíciles. A los que formé en la universidad, quienes compartieron conmigo este desafío académico, convirtiéndose en una segunda familia.

A Bryan Pineda, Jean Pierre Mera, Joseph Ríos, Santiago Tene, Omar Maldonado, Adrián Núñez y Luis Jiménez, por su apoyo incondicional y por acompañarme en este camino.

A mi mejor amigo y compañero de trabajo, Andrés Macas, por su amistad sincera, por compartir conmigo el esfuerzo, las largas jornadas de trabajo y el compromiso que nos ha llevado hasta aquí. Gracias por ser un compañero excepcional en esta etapa.

A mi tutor de Trabajo de Integración Curricular, Andrés Navas, por su guía, paciencia y apoyo durante todo este proceso. Su orientación ha sido fundamental para alcanzar esta meta.

Finalmente, a todas las personas que, de una u otra manera, formaron parte de mi vida durante este camino, sigan estando presentes o no. Gracias por sus enseñanzas y lecciones, que han impulsado mi crecimiento tanto personal como académico.

A todos ustedes, mi gratitud infinita.

Bryan Jeremy Encalada Mejía

Índice de Contenidos

Portada	i
Certificación de director	ii
Autoría	iii
Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Integración Curricular ..	iv
Dedicatoria	v
Agradecimiento	vii
Índice de Contenidos	ix
1. Título	1
2. Resumen	2
Abstract	3
3. Introducción	4
4. Marco teórico	6
4.1. Antecedentes	6
4.2. Realidad Virtual.....	6
4.2.1. Realidad Virtual no Inmersiva	7
4.2.2. Realidad Virtual Semi-Inmersiva.....	7
4.2.3. Realidad Virtual Inmersiva	7
4.2.4. Metodología de desarrollo para entornos de realidad virtual	9
4.2.5. Medición de sensación de presencia percibida.....	10
4.3. Tecnologías y herramientas para el desarrollo de realidades virtuales inmersivas ..	11
4.3.1. HMD	11
4.3.2. Mixamo	12
4.3.3. C#.....	12
4.3.4. Visual Studio.....	12
4.3.5. Visual Studio Code	13
4.3.6. Motores gráficos	13
4.3.7. Herramientas de modelado 3D	14
4.4. Inteligencia artificial generativa.....	15
4.4.1. Modelo Google Gemini 1.5 Pro.....	15
4.4.2. HuggingFace	15
4.4.3. Azure	16
4.4.4. Backend	16

4.5.	Herramientas Complementarias.....	17
4.5.1.	Adobe Photoshop	17
4.5.2.	Git.....	17
4.5.3.	GitHub	18
4.5.4.	Git Large File Storage (LFS).....	18
4.6.	Trabajos Relacionados.....	18
5.	Metodología	20
5.1.	Área de estudio	20
5.2.	Contexto.....	20
5.3.	Procedimiento	21
5.3.1.	Objetivo específico 1: Actualizar el sistema de tour virtual de la carrera Computación de la FEIRNNR a un entorno inmersivo mediante la metodología MSD-VR. 21	
5.3.1.1.	Fase de asignación.....	21
5.3.1.2.	Fase de análisis.....	21
5.3.1.3.	Fase de construcción.....	22
5.3.1.4.	Fase de implementación.....	22
5.3.1.5.	Fase de pruebas.....	24
5.3.2.	Objetivo específico 2: Implementar un sistema de respuestas inteligentes y proactivas, con el modelo generativo "Gemini 1.5 Pro", para los avatares informativos dentro del sistema virtual inmersiva.....	25
5.3.2.1.	Fase de asignación.....	25
5.3.2.2.	Fase de análisis.....	26
5.3.2.3.	Fase de construcción.....	26
5.3.2.4.	Fase de implementación.....	27
5.3.2.5.	Fase de pruebas.....	28
6.	Resultados.....	30
6.1.	Objetivo 1: Actualizar el sistema de tour virtual de la carrera Computación de la FEIRNNR a un entorno inmersivo mediante la metodología MSD-VR.	30
6.2.	Objetivo 2: Implementar un sistema de respuestas inteligentes y proactivas, con el modelo generativo "Gemini 1.5 Pro", para los avatares informativos dentro del sistema virtual inmersiva.....	51
7.	Discusión.....	69
7.1	Primer objetivo:.....	69
7.2	Segundo objetivo	70
8.	Conclusiones.....	72
9.	Recomendaciones.....	73

10. Anexos	74
11. Bibliografía	88

Índice de Tablas

Tabla 1. Tests aplicados entre un entorno inmersivo y uno no inmersivo [14].....	9
Tabla 2. Ventajas de usar Visual Studio como IDE del entorno .NET.....	13
Tabla 3. Comparación de motores gráficos.....	13
Tabla 4. Comparación de herramientas de modelado 3D	14
Tabla 5. Comparación de frameworks Backend	17
Tabla 6. Resumen de trabajos relacionados	18
Tabla 7. Resumen de uso de aplicaciones de Meta para el desarrollo de aplicaciones para Meta Quest 2.....	22
Tabla 8. <i>Requisitos funcionales del tour virtual</i>	30
Tabla 9. Requisitos no funcionales del tour virtual	31
Tabla 10. Escenarios a recrear dentro del tour virtual	32
Tabla 11. Clases y atributos del diagrama	34
Tabla 12. Descripción de técnicas de movimiento.....	42
Tabla 13. Resultado de las pruebas de usuario	50
Tabla 14. Descripción y función de cada archivo del BackEnd	52
Tabla 15. Resultado de aplicación de test SuS	66
Tabla 16. Resultado de SoP en entorno no inmersivo y entorno inmersivo.....	68

Índice de figuras

Figura 1. Fases de la metodología MSD-VR [9].....	9
Figura 2. Proceso de desempaquetado	14
Figura 3. Mapa de la secretaría de la carrera Computación de la UNL.....	20
Figura 4. Dispositivos Meta Quest 2 conectados exitosamente a Meta Quest Link.....	23
Figura 5. Conexión exitosa de Meta Quest 2 a Meta Developer Hub.....	23
Figura 6. Configuración de Unity para exportar el entorno inmersivo.....	24
Figura 7. Control derecho de dispositivo Meta Quest 2 [44].....	27
Figura 8. Diagrama de clases del entorno virtual inmersivo	34
Figura 9. Diagrama de componentes de entorno virtual inmersivo (Elaboración propia).....	35
Figura 10. Planos del edificio de secretaría de la facultad (Elaboración propia).....	36
Figura 11. Ruta de archivos de proyecto del edificio en Blender.....	37
Figura 12. Modelado de paredes exteriores del edificio de secretaría	38
Figura 13. Modelado de paredes interiores del edificio de secretaría	39
Figura 14. Modelado del piso del edificio de secretaría	40
Figura 15. Modelado del techo del edificio de secretaría	41
Figura 16. Modelado final del edificio de secretaría en Blender	42
Figura 17. Script para la integración de Arm Swing	45
Figura 18. Importe del modelo de edificio junto al SkyBox dentro de Unity	46
Figura 19. Integración de edificio y elementos exteriores del edificio de secretaría dentro de Unity.....	47
Figura 20. Configuración para exportar entorno virtual a Meta Quest 2	48
Figura 21. Testeo de APK exportado e importado a los Meta Quest 2.....	49
Figura 22. Diagrama de procesos.....	51
Figura 23. Estructura de la ruta de archivos para el BackEnd.....	52
Figura 24. Ejemplo de formato de respuesta a las solicitudes en JSON.	53
Figura 25. Cuenta creada y validada de Hugging Face	54
Figura 26. Configuración de la biblioteca de HuggingFace dentro de Unity	55
Figura 27. Script de Unity en C# para los métodos IniciarReconocimientoVoz y StartRecording	56
Figura 28. Script de Unity en C# para los métodos StopRecording y SendRecording	58
Figura 29. Configuración del servicio Azure Speech.....	58
Figura 30. Métodos LoadAudioFromFile y SendTextToSpeechRequest.....	61
Figura 31. Modelo 3D seleccionado en Mixamo	61
Figura 32. Collider de Unity para la detección del jugador	62
Figura 33. Configuración de Collider dentro de Unity.....	62
Figura 34. Método OnTriggerEnter para detectar la presencia del jugador	63

Figura 35. Código de Método StartRecording y asignación a botón de controles físicos	64
Figura 36. Código para realizar la solicitud	65
Figura 37. Representación gráfica de los resultados comparativos en la SoP percibida en ambos entornos.....	70
Figura 38. Evidencia 1 - Exposición y aplicación de encuestas (Quinto Ciclo)	86
Figura 39. Evidencia 2 - Exposición y aplicación de encuestas (Primer Ciclo)	87

Índice de Anexos

Anexo 1. Repositorio de código fuente del Sistema Virtual Inmersivo.....	74
Anexo 2. Transcripción de entrevistas técnicas para la elicitación de requisitos funcionales y no funcionales.	75
Anexo 3. Evidencia de exposición del TIC en "VR Day" en la UNACH.	78
Anexo 4. Repositorio de código fuente del Backend del Sistema Virtual Inmersivo	79
Anexo 5. Cuestionario de sensación de presencia percibida	80
Anexo 6. Resultado de aplicación de encuestas a 39 estudiantes	81
Anexo 7. Certificado de culminación de TIC	85
Anexo 8. Evidencia de aplicación de encuestas a estudiantes de la UNL.....	86
Anexo 9. Certificado de traducción Resumen - Abstract.....	88

1. Título

Desarrollo de realidad virtual inmersiva aplicada al entorno virtual de la carrera Computación e integración de respuestas inteligentes mediante la implementación de Inteligencia Artificial

Development of Immersive Virtual Reality for the Computer Science Degree's Virtual Environment and Integration of Intelligent Responses Through Artificial Intelligence

2. Resumen

El presente trabajo de integración curricular aborda la actualización del sistema de tour virtual de la Carrera de Computación de la Universidad Nacional de Loja, transformándolo en un entorno completamente inmersivo mediante el uso de la realidad virtual y la inteligencia artificial. Se empleó la metodología MSD-VR para estructurar el desarrollo del sistema, permitiendo optimizar su implementación y ejecución en los dispositivos Meta Quest 2. Además, se incorporó el modelo generativo "Gemini 1.5 Pro" para la creación de avatares informativos con respuestas inteligentes en tiempo real, mejorando la interacción del usuario dentro del entorno virtual con el objetivo de aumentar la inmersión.

Los resultados de este estudio evidencian una mejora significativa en la sensación de presencia percibida de la muestra seleccionada para la aplicación del test SUS, en comparación con el sistema de tour virtual no inmersivo previo. Se logró incrementar la sensación de inmersión gracias a la optimización del modelado 3D, la interacción fluida, la integración de texturas de buena calidad y técnicas de desplazamiento realistas dentro del entorno. Sin embargo, se identificaron desafíos relacionados con la optimización de los recursos computacionales y la adaptabilidad de la metodología MSD-VR a proyectos educativos específicos.

Las conclusiones del estudio destacan la efectividad de la realidad virtual inmersiva como herramienta de enseñanza y orientación académica. Asimismo, se recomienda la implementación de futuras mejoras, como la optimización de la inteligencia artificial y la exploración de nuevas tecnologías de interacción. Este trabajo representa un avance en la aplicación de entornos virtuales inmersivos en el ámbito de tours virtuales, sentando las bases para su adopción en diferentes áreas.

Palabras clave: Sensación de presencia, realidad virtual, inmersión, inteligencia artificial, entorno inmersivo.

Abstract

This degree work deals with the updating of the virtual tour system of Carrera de Computación of Universidad Nacional de Loja, transforming it into a fully immersive environment through the use of virtual reality and artificial intelligence. The MSD-VR methodology was used to structure the development of the system, allowing to optimize its implementation and execution in Meta Quest 2 devices. In addition, the generative model “Gemini 1.5 Pro” was incorporated for the creation of informative avatars with intelligent responses in real time, improving user interaction within the virtual environment in order to increase immersion.

The results of this study show a significant improvement in the perceived sense of presence of the sample selected for the application of the SUS test, compared to the previous non-immersive virtual tour system. An increased sense of immersion was achieved due to optimized 3D modeling, smooth interaction, integration of good quality textures and realistic scrolling techniques within the environment. However, challenges related to the optimization of computational resources and the adaptability of the MSD-VR methodology to specific educational projects were identified.

The conclusions of the study highlight the effectiveness of immersive virtual reality as a teaching and academic orientation tool. It also recommends the implementation of future improvements, such as the optimization of artificial intelligence and the exploration of new interaction technologies. This work represents an advance in the application of immersive virtual environments in the field of virtual tours, laying the groundwork for its adoption in different areas.

Keywords: Sense of presence, virtual reality, immersion, artificial intelligence, immersive environment.

3. Introducción

El avance de la tecnología ha permitido el desarrollo de entornos virtuales cada vez más realistas, capaces de proporcionar experiencias inmersivas en distintos campos del conocimiento. En particular, la realidad virtual inmersiva ha cobrado relevancia en el ámbito académico, al facilitar la representación de espacios físicos mediante entornos simulados que potencian la interacción y la sensación de presencia de los usuarios. Estas tecnologías se han implementado en diversas disciplinas, incluyendo la educación, la medicina y la ingeniería, permitiendo a estudiantes y profesionales interactuar con modelos tridimensionales y escenarios simulados que enriquecen el aprendizaje y el entrenamiento práctico.

En la actualidad, la Carrera de Computación de la Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables (FEIRNNR) de la Universidad Nacional de Loja (UNL) cuenta con un sistema de tour virtual no inmersivo que permite a los usuarios explorar de manera digital el edificio administrativo de la carrera. Con el surgimiento de la realidad virtual, surge la interrogante de si la implementación de entornos completamente inmersivos mejora la sensación de presencia percibida, ofreciendo una experiencia más natural y envolvente. Este enfoque propone que, al migrar hacia sistemas que permiten desplazarse e interactuar en entornos tridimensionales, se puede potenciar la percepción espacial y la conexión del usuario con el entorno, abriendo nuevas posibilidades para la representación y el acceso a espacios educativos.

Con el fin de explorar esta posibilidad, el presente Trabajo de Integración Curricular propone la actualización del sistema de tour virtual hacia un entorno completamente inmersivo, mediante la implementación de tecnologías de realidad virtual y el uso de dispositivos especializados como las gafas Meta Quest 2. La experiencia inmersiva ha demostrado ser una herramienta poderosa para mejorar la retención de información y la comprensión espacial, aspectos clave en contextos educativos. Al permitir que los usuarios se desplacen e interactúen de manera natural en un espacio tridimensional, se fomenta un aprendizaje más intuitivo y significativo.

Para llevar a cabo esta actualización, se emplea la metodología MSD-VR, la cual estructura el desarrollo en diversas fases que garantizan una implementación efectiva del sistema. Este enfoque iterativo permite optimizar cada etapa del desarrollo, asegurando que el producto final cumpla con estándares requeridos. La metodología facilita la integración de diversas tecnologías, tales como motores gráficos avanzados, técnicas de modelado 3D y

sistemas de interacción por voz, lo que contribuye a crear una experiencia más envolvente y natural para los usuarios.

Adicionalmente, se plantea la incorporación de un sistema de respuestas inteligentes basado en el modelo generativo "Gemini 1.5 Pro". Este sistema permitirá una interacción natural entre los usuarios y los avatares informativos dentro del entorno virtual, utilizando inteligencia artificial para mejorar la experiencia de navegación y la percepción de presencia. La implementación de esta tecnología no solo enriquece la interacción, sino que también facilita el acceso dinámico y personalizado a la información, permitiendo que los avatares respondan en tiempo real y guíen a los usuarios a través de los distintos espacios y servicios de la facultad.

A lo largo de este documento, se abordará el marco teórico que sustenta la implementación del entorno virtual inmersivo, así como las tecnologías y herramientas empleadas para su desarrollo. Se detallará el proceso de selección de hardware y software, incluyendo una comparación de diferentes motores gráficos y dispositivos de realidad virtual, con el fin de asegurar la mejor experiencia posible. Asimismo, se describirá la metodología utilizada, los resultados obtenidos y el impacto que la migración hacia un entorno inmersivo tiene en la percepción de presencia de los usuarios. Finalmente, se presentarán las conclusiones y recomendaciones derivadas del estudio, con el objetivo de contribuir al desarrollo de entornos virtuales educativos más efectivos y envolventes.

Este trabajo busca demostrar cómo la implementación de tecnologías avanzadas de realidad virtual e inteligencia artificial puede transformar la manera en que los usuarios interactúan con los espacios educativos digitales, cuestionando y explorando si el entorno inmersivo incrementa la sensación de presencia y mejora la experiencia educativa. Se espera que los resultados de este estudio sirvan como referencia para futuras iniciativas de modernización en la enseñanza y promoción académica, fomentando el uso de herramientas innovadoras para enriquecer la experiencia educativa y administrativa dentro de la Universidad Nacional de Loja.

4. Marco teórico

4.1. Antecedentes

Hasta el año 2023, la carrera de Computación de la Facultad de Energía, Industrias y Recursos Naturales Renovables (FEIRNNR) de la Universidad Nacional de Loja (En adelante UNL), ubicada en Loja-Ecuador, no disponía de la herramienta de un tour virtual para su facultad. Eventualmente, a través de un trabajo de integración curricular, se implementó un tour de realidad virtual, este proyecto preliminar “Energía VR” fue desarrollado por el autor Pucha J. [1].

El sistema virtual desarrollado por Pucha J. [1] permitió evaluar la percepción de inmersión en entornos virtuales no inmersivos. El sistema fue sometido a pruebas, y los resultados arrojaron un puntaje de 6,01 en la escala de Likert (1-7), situándose cerca del valor máximo. Sin embargo, el entorno puede ser mejorado para incrementar esta percepción. Factores como texturas de baja o media calidad, animaciones rígidas, movimientos poco naturales de los avatares y otros elementos externos limitan la experiencia [2], a pesar de su alta puntuación. Por lo tanto, tras analizar la bibliografía y los resultados de las encuestas realizadas al personal técnico de la carrera de computación, se concluye que el sistema requiere una actualización que integre no solo tecnologías de realidad inmersiva, sino también dispositivos como gafas o controles de realidad virtual (Meta Quest 2). Estas mejoras transformarán el sistema de tour virtual en una experiencia plenamente inmersiva, cuyo impacto será validado mediante cuestionarios de percepción de inmersión (SUS).

4.2. Realidad Virtual

La definición de realidad virtual ha ido evolucionando al igual que la tecnología con el pasar de los años [2], y por ende, ha ido evolucionando, en el año 2023 los autores Jianghao, En-Lin, Ziqian, Tao, Shin [3], la definieron como la interacción de un humano y un computador mediante el uso de una interfaz avanzada.

El avance tecnológico ha permitido el desarrollo de distintos tipos de realidad virtual, basados en su nivel de inmersión [4]. El más básico es el no inmersivo, que utiliza un monitor que funciona como una ventana al mundo virtual [3]. Dentro de la realidad virtual, los usuarios podrán moverse dentro del mundo virtual 3D y verlo desde distintos ángulos. La diferencia entre los entornos virtuales y los mundos virtuales es que, este último término se ha aplicado a espacios sociales en línea persistentes; es decir, entornos virtuales que las personas experimentan, junto con otros, como un mundo para la interacción social [3]. Un entorno virtual simula un espacio reducido que debe ser simulado, normalmente con fines informativos y educativos.

4.2.1. Realidad Virtual no Inmersiva

Se entiende por realidad virtual no inmersiva aquellas experiencias que muestran el contenido virtual en una pantalla convencional de una computadora y cuya interacción con este entorno se realiza a través de diferentes componentes como un teclado, mouse, micrófono, audífonos, joysticks, etc.[4], [5]

Este tipo de inmersividad es ampliamente utilizado en diferentes áreas del mercado actual, especialmente dentro del área del entretenimiento, donde entran campos como los videojuegos o el turismo, además recientemente se ha logrado implementar en campos más científicos como la medicina, donde se han podido generar simulaciones de partes del cuerpo humano ya sea con propósitos prácticos o académicos, debido principalmente a que su costo tanto económico como de potencia computacional no suele ser tan exigente.[4], [6]

4.2.2. Realidad Virtual Semi-Inmersiva

La realidad virtual semi-inmersiva puede interpretarse como una mezcla entre ambos extremos del tipo de inmersividad, para esto se utilizan pantallas convencionales de un computador, pero se encuentran colocadas de tal manera que permiten abarcar todo el campo de vista del usuario, por lo mismo estas pantallas se encuentran normalmente en las paredes y en ocasiones especiales en el suelo, de igual forma los controles para interactuar con estos sistemas suelen ser poco convencionales. Esta tecnología suele usar con mayor medida dentro del área de entretenimiento, sobre todo para simular experiencias como montañas rusas o conducción de vehículos y normalmente no suelen tener un coste computacional muy elevado, pero sí un alto coste económico.[4], [6], [7]

4.2.3. Realidad Virtual Inmersiva

Dentro de la informática, la inmersión, definida por los autores Bailenson y Bailey [8], es la extensión en la que un sistema computacional puede crear entornos que simulan bien al mundo físico. Para esto, se hace uso de las modalidades sensoriales del usuario, con el objetivo de crear una experiencia más vívida.

También, similar al concepto brindado para la realidad virtual más tradicional, la inmersiva involucra la interacción usuario computador, pero de una forma más profunda con el entorno virtual [2], haciendo uso de herramientas y tecnologías avanzadas relacionadas a la percepción sensorial. Este tipo de herramientas VR pueden ir desde cascos de realidad virtual, controles que simulan las manos, sistemas de rastreo de movimiento, controladores para uso del espacio virtual [9], ayudando así a minimizar la sensación que impone ver un mundo digital desde una pantalla [10].

Una de las características fundamentales de la realidad virtual inmersiva es la creación de una fuerte sensación de presencia, lo que significa que los usuarios sienten que están realmente dentro del entorno virtual. Para lograr esto, se utilizan técnicas avanzadas de renderizado gráfico y procesamiento de sonido espacial que imitan la forma en que percibimos el mundo físico, lo que aumenta la percepción de realismo. Además, la interacción en estos entornos no se limita a la vista o el oído, sino que puede involucrar otros sentidos como el tacto, mediante dispositivos que simulan sensaciones físicas como el contacto o la resistencia [3], [11].

La realidad virtual inmersiva ha encontrado aplicaciones en campos que requieren simulaciones realistas, como la capacitación militar [5], la simulación de desastres naturales [6] y la enseñanza en ciencias de la salud. En estos contextos, la inmersión total permite a los usuarios experimentar y aprender en situaciones cercanas a la realidad sin enfrentarse a los peligros que supondría en el mundo físico [7]. Sin embargo, esta modalidad aún presenta desafíos importantes, como los altos costos de implementación y la necesidad de un hardware especializado, lo que puede limitar su adopción en ciertos entornos educativos o profesionales. A medida que las tecnologías de VR inmersiva continúan evolucionando, se espera que estas barreras disminuyan y que los entornos virtuales sean más accesibles y realistas.

- **Sensación de presencia e inmersión**

Según el análisis realizado en un artículo [9], la inmersión se define como aquella percepción o sensación que tienen los jugadores o usuarios cuando se sienten “presentes” dentro del entorno virtual. Por otra parte, los investigadores Lénaïc y Chainay [12], definen la sensación de presencia (en adelante SoP según la terminología Sense of Presence) como un aspecto relacionado de manera directa con la telepresencia, lo que quiere decir que se basa en qué medida los usuarios responden a la sensación de estar presentes dentro del entorno virtual.

Además, tomando en cuenta los análisis realizados de la información bibliográfica [12], [13], se infiere que un entorno de realidad virtual inmersivo debería tener una SoP de mayor medida en comparación a un entorno que no tenga características inmersivas. En la **Tabla 1** se puede observar el resultado de un estudio realizado en el año 2023 por la Universidad Norwegian [14], donde, se realizaron pruebas en grupos de estudiantes que interactuaron con dos ambientes de realidad virtual, uno de escritorio (o tradicional) y otro inmersivo. Se evidenció que la SoP percibida es mayor en un entorno inmersivo (7.10) a diferencia del entorno de escritorio (5.77).

Tabla 1.

Tests aplicados entre un entorno inmersivo y uno no inmersivo [14]

	<i>Immersive VR</i>	<i>Desktop VR</i>
	<i>M</i>	<i>M</i>
Sense of presence	7.10	5.77

Nota: Media (M), Realidad Virtual (VR)

- **Limitaciones**

No es de desconocimiento que la tecnología, y por ende la realidad virtual, se encuentra en constante cambio y actualización, y los campos en los que puede ser aplicada esta tecnología pueden no cubrir las necesidades [12]. Estos campos pueden ir desde la formación estudiantil médica, avances científicos, históricos, lingüísticos, artísticos, etc. Y la capacidad de un computador para generar un ambiente 3D es cada vez más exigente.

4.2.4. Metodología de desarrollo para entornos de realidad virtual

La elección de una metodología de desarrollo de software dentro de un proyecto tecnológico es clave, ya que influye directamente en todo el ciclo de vida del proyecto [7]. Una metodología bien seleccionada no solo define las fases en las que se dividirá el proyecto, sino que también establece cómo se abordará cada una de estas etapas, lo que puede impactar significativamente en el éxito o fracaso del mismo [8].

Dentro del mundo del desarrollo de realidad virtual inmersiva, se encuentra la metodología MSD-VR (Methodology for Designing Virtual Reality Applications)[9], que propone un proceso estructurado y detallado para la creación de aplicaciones de realidad virtual. Este enfoque abarca de manera sistemática y técnica cada una de las fases del desarrollo, desde la comunicación inicial con el cliente hasta la implementación y pruebas finales. La **Figura 1** muestra las siete fases principales de esta metodología, siendo la fase de asignación, de análisis, de creación, de pruebas, de implementación y de operación.



Figura 1.

Fases de la metodología MSD-VR [9]

Esta metodología tiene un enfoque especializado en el desarrollo de aplicaciones de realidad virtual, así como en su capacidad para integrarse de manera efectiva con la implementación de respuestas interactivas dentro del entorno.

4.2.5. Medición de sensación de presencia percibida

Históricamente han existido varios enfoques de cuestionarios para medir la SoP percibida, uno de estos cuestionarios fue desarrollado por Slater y sus colegas en 1998, y fue denominado SUS (Slater-Usuh-Steed) [15]. A pesar de existir otros tipos de cuestionarios para medir SoP, como el planteado por Igroup (IPQ) [10] o el planteado por Witmer and Singer (WS) [11], se optó por SUS debido a que las preguntas planteadas por este cuestionario se ajustan a los tres temas principales del TIC: la sensación de estar dentro del entorno virtual, el grado en el que el entorno virtual tiene una realidad dominante y el grado en el que el entorno virtual es recordado como un lugar físico [16].

- **Escala de Likert**

La escala de medición creada por el psicólogo Rensis Likert permite la medición de opiniones y aptitudes mediante una escala de 5 a 7 puntos que normalmente se detallan de una puntuación baja (En desacuerdo o 1) a una alta (En acuerdo o 7) [12], [1], [13]. Por otro lado, luego de un análisis del Alfa de Cronbach se concluye que se recomienda usar la escala de Likert en una escala de 7 puntos para obtener resultados más fiables [13].

Para nuestro estudio, y siguiendo la metodología de Pucha, la forma de medir los resultados va de escala del 1 a 7, donde 1 es una SoP muy baja y 7 es muy alta. La fórmula utilizada para el cálculo varía en función del número de preguntas (n) planteadas y el número de respuestas (r) cuya puntuación está dentro del rango de 6 o 7 [1].

- **Alfa de Cronbach**

El alfa de Cronbach fue propuesta por Lee Cronbach en 1951, y es utilizado para medir el nivel de confiabilidad de los resultados de aplicar un instrumento como una encuesta o cuestionario. Su cálculo parte de la varianza individual de cada item, pero, si la escala de todos es la misma se toma en cuenta también la suma de la varianza de los items de cada usuario que participe en el instrumento [14], [15].

El cálculo se encuentra delimitado en un rango que va desde -1 hasta 1, cuando el resultado tiende a valores negativos indica que los resultados no son tan fiables. Por otro lado,

si se obtiene un cálculo de 0.65 a 0.80 se considera adecuado dentro del campo de la psicología [14], [15].

- **Cuestionario SUS aplicado**

Debido a que la pregunta de investigación del TIC está orientada a ser respondida mediante la actualización del entorno virtual creado por Pucha [1], el cuestionario para medir la SoP percibida es similar. Este test consta de 6 ítems, relacionados con los propuestos por Slater [16].

Gracias a diversos estudios [17]., [2], se conoce que los estímulos externos como la audición, visión y movimiento afectan de forma directa en la inmersión y SoP, el cuestionario se enfoca en medir la "sensación de allí" y la "ilusión de plausibilidad", que es la percepción que crea el usuario al sentir si la experiencia virtual está sucediendo de verdad.

El cuestionario fue adaptado al nuevo contexto del entorno virtual inmersivo, por lo que, algunas preguntas fueron reformuladas y mejoradas en redacción para su mejor entendimiento en los estudiantes encuestados. La escala final elegida es la misma de la del test aplicado por Pucha, que va desde 1 (De ninguna manera) hasta 7 (Mucho) (Véase el cuestionario completo en la sección **Anexo 5**).

4.3. Tecnologías y herramientas para el desarrollo de realidades virtuales inmersivas

Elegir las tecnologías adecuadas es quizás uno de los aspectos clave en cualquier proyecto, sobre todo en proyectos de software pues de esto dependerá cómo será la experiencia de desarrollo del equipo y a su vez qué tan fácil será lidiar con errores, documentación y demás aspectos [18]. Esta importancia aumenta mucho más cuando se trata de un proyecto de realidad virtual inmersiva, o en este caso, de un tour virtual, donde la elección puede contribuir o deshacer el éxito del mismo [19]. Por esto, se debe optar por una selección detallada y precisa de herramientas para asegurar la elección de las mismas y de los métodos adecuados que se ajusten a al propósito de la realización del proyecto, todo esto con el fin de establecer una experiencia de desarrollo que no presente problemas técnicos en sus herramientas y así no sólo que sea una fase eficiente sino también sólida, priorizando por encima de todo la satisfacción de los usuarios finales.

4.3.1. HMD

Head-Mounted Display, por sus siglas HMD, es un dispositivo físico que se coloca en la cabeza del usuario al igual que un casco y que posee un display o pantalla en el cual se transmiten imágenes de manera muy aproximada a la vista generando así un campo de visión

que cubre toda la vista del usuario y permite generar simulaciones más inmersivas para el mismo [17], [18].

- **Meta Quest 2**

A la fecha, Meta Quest 2¹ es uno de los dispositivos de realidad virtual inmersiva más populares. El casco desarrollado por Meta permite a los usuarios adentrarse dentro de entornos virtuales simulados [19].

4.3.2. Mixamo

Mixamo² es una herramienta de Adobe que brinda una solución a la complejidad de la animación 3D de personajes a cuerpo completo [20]. Permite a los usuarios diseñar, crear y personalizar personajes 3D para varios fines como juegos, películas, animaciones y otras experiencias interactivas [21].

Además de permitir la personalización de personajes, Mixamo tiene una capa de animaciones predeterminadas, que pueden ser exportadas junto con el modelo elegido por el usuario [21]. Gracias a esto, se simplifica el proceso de animación 3D, evitando el uso de tecnologías complejas y costosas como motion capture o rigging automático. Además, permite exportar el personaje en distintos formatos compatibles con varios entornos de desarrollo 3D y motores gráficos [22].

4.3.3. C#

C# es un lenguaje de programación desarrollado por Microsoft, forma parte de la plataforma .NET [23]. Al ser desarrollado para una plataforma única, los códigos escritos en C# pueden referenciar a otro escrito en el entorno de VB.NET [24].

Motores gráficos como Unity utilizan a C# como su lenguaje de programación de scripts, con él, se pueden automatizar eventos, o detectar acciones para generar respuestas (físicas, animaciones, texto, voz, etc.). Además, permite utilizar los paquetes creados por Unity para interactuar de mejor manera con controles de dispositivos especiales [25].

4.3.4. Visual Studio

Visual Studio³ es un entorno de desarrollo integrado o IDE, está desarrollado por Microsoft y está enfocado al desarrollo de aplicaciones del entorno .NET. Este IDE ofrece

¹ Meta Quest 2: <https://www.meta.com/es/quest/products/quest-2/>

² Mixamo: <https://www.mixamo.com>

³ Visual Studio IDE: <https://visualstudio.microsoft.com/es/>

herramientas y funciones que lo hacen en una opción óptima para desarrollar scripts y software complejo, la **Tabla 2** resume las ventajas que tiene el uso de este IDE [26].

Tabla 2.

Ventajas de usar Visual Studio como IDE del entorno .NET

Característica	Descripción
Soporte de lenguajes	Visual Studio soporta lenguajes del entorno de Microsoft como C#, sin embargo, también permite el desarrollo de código en C++, Python, JavaScript, etc.
Depuración	Brinda herramientas de depuración avanzada que permiten identificar errores de manera fácil y ágil, lo que acelera el proceso de desarrollo.
IntelliSense	Ofrece un sistema de autocompletado de código, nombres, métodos, propiedades, etc.
Integración con IA	Actualmente, Visual Studio tiene soporte para GitHub Copilot, que es un asistente de IA que permite la generación de código y funciones.
Integración con Git	Tiene soporte nativo para el control de versiones de Git.
Extensiones	Amplia gama de extensiones para personalizar y adaptar el Ide a las necesidades de desarrollo.

Información extraída del sitio oficial de Visual Studio³

4.3.5. Visual Studio Code

Visual Studio Code⁴ o VS Code, es un editor de código popular en el mundo del desarrollo de software. Desarrollado por Microsoft, es una herramienta utilizada, sobre todo en entornos de desarrolladores que usan JavaScript, Python, etc. Gracias a su flexibilidad y alto rendimiento lo hacen ideal para la construcción de APIs o para integrarse con frameworks (gracias a su amplia gama de extensiones de todo tipo).

Esta herramienta permite integrarse con Node.JS y Express para desarrollar un intermediario que facilita la comunicación con el API de Google Gemini.

4.3.6. Motores gráficos

La **Tabla 3** muestra un análisis comparativo sobre los motores gráficos más comunes, con el objetivo de analizar ventajas y desventajas entre cada uno.

Tabla 3.

Comparación de motores gráficos.

Característica	Unity	Unreal Engine	Godot
Lenguajes	C#, Visual Scripting	C++, Blueprint	GScript, C#, Visual Script
Licencia	Disponibilidad de licencia gratuita limitada	Gratis hasta \$1.000.000 de ingresos	Gratis
Documentación	Amplia	Amplia	Aún en desarrollo
Gráficos	Útil para entornos 3D y 2D	Orientado a entornos 3D	Útil para entornos 3D y 2D flexibles

Información extraída de sitios oficiales de Unity [25], Unreal Engine [27] y Godot [28].

⁴ Visual Studio Code IDE: <https://code.visualstudio.com/>

4.3.7. Herramientas de modelado 3D

La **Tabla 4** presenta el análisis comparativo de los distintos motores gráficos existentes.

Tabla 4.
Comparación de herramientas de modelado 3D

Característica	Blender	Maya	3ds Max
Licencia	Gratuita (Código abierto)	Pagada	Pagada
Facilidad de uso	Alta	Compleja	Intermedia
Documentación	Amplia y activa	Amplia	Amplia en ámbito de arquitectura
Integración	Fácil con Unity y Unreal Engine	Centrada en integración con AutoDesk	Centrada en integración con AutoDesk
Características	Animación, modelado, simulaciones, mapeo de texturas y materiales, etc.	Animación avanzada, Rigging profesional.	Modelado arquitectónico y simulaciones simples.

Información extraída de sitios oficiales de Blender [29], Maya [30] y 3ds Max [31].

- **UV Mapping**

Dentro del mundo del modelado 2D, se utiliza un sistema de coordenadas X, Y para describir la superficie del entorno. En cambio, en el mundo del modelado 3D, se usa un entorno de tres coordenadas X, Y y Z. Para definir mejor este sistema de coordenadas 3D se denomina con las letras U y V. En este caso, para Z se asigna W, aunque en herramientas de modelado como Blender no es utilizado [24].

El concepto radica en realizar un mapeo UV entre la superficie 2D de una imagen con herramientas específicas de cada software de modelado. Con estas herramientas se puede realizar el proceso de UV Mapping. La **Figura 2** muestra parte del proceso de esta técnica, que consiste en realizar cortes manuales para desplegar un plano y poder texturizarlo en un editor UV [25].

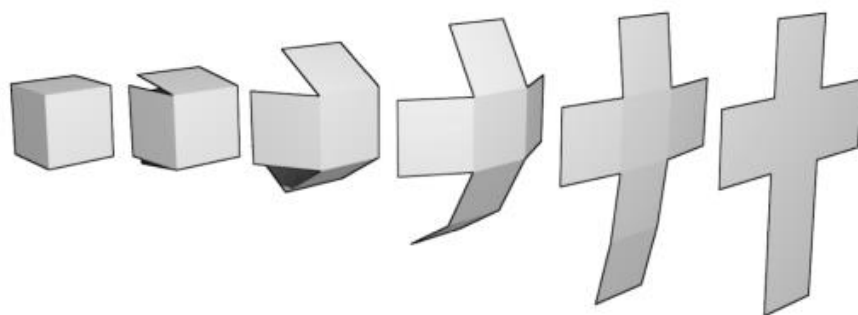


Figura 2.
Proceso de desempaquetado

4.4. Inteligencia artificial generativa

Se denomina inteligencia artificial a aquel producto informático que busca generar un resultado lo más cercano a como lo haría un humano [26]. Por otro lado, una inteligencia artificial generativa de texto es un modelo que puede generar contenido textual mediante el análisis de entradas de un usuario. Los chatbots buscan simular lo más cercano a una conversación real con un humano, y existen distintos modelos que generan texto con estas características mediante inteligencia artificial como ChatGPT, Dalle-2, etc. [27].

El contenido generado por una IA es resultado del aprendizaje y análisis de datos existentes en sus sistemas, donde guardan no solo el conocimiento, sino que un motor que permite ajustar las reglas para generar el contenido [28].

4.4.1. Modelo Google Gemini 1.5 Pro

Gemini es uno de los varios chatbots basados en inteligencia artificial que se encuentran en el mercado actual, pero este posee la ventaja de tener un api cuya capa gratuita permite su uso de manera poco limitada, específicamente la versión Gemini 1.5 Pro es un modelo avanzado de procesamiento del lenguaje natural (NLP) que permite comprender y representar texto de forma coherente y natural. Gemini posee una gran capacidad para mejorar la interacción del usuario con entornos virtuales, proporcionar contenido relevante y ofrecer respuestas atractivas y con un contexto establecido. El uso de Gemini en aplicaciones de realidad virtual requiere la creación de una interfaz que permita la creación y control de estas respuestas inteligentes, combinando así la inteligencia artificial con otros entornos virtuales.[29]

4.4.2. HuggingFace

HuggingFace es una plataforma o sitio web enfocado en los modelos de inteligencia artificial, específicamente en los modelos de procesamiento de lenguaje natural (por sus siglas NLP) visión por lenguaje y audio, de igual forma también ofrece diferentes herramientas como Datasets, Transformers, conversores de texto a audio y demás, facilitando de esta manera el acceso a diferentes modelos pre-entrenados y a su vez permitiendo la posibilidad de poder desplegarlos de manera local. No obstante, HuggingFace también ofrece una API para poder interactuar con estos modelos sin necesidad de ejecutarlos en el computador del usuario, si no que interactuando con ellos a través de consultas HTTP para así poder acceder a las diferentes funcionalidades que ofrece este servicio [32].

4.4.3. Azure

Es el servicio de computación en la nube ofertado por Microsoft, este tiene una amplia variedad de servicios o funcionalidades tanto para el ámbito de desarrollo, despliegue, infraestructura, inteligencia artificial y almacenamiento, además de permitir una escalabilidad flexible y unos costes de servicio basados en el uso que se le dé a cada servicio y no tanto a una mensualidad fija [33].

- **Azure Speech**

Azure Speech es uno de los tantos servicios que ofrece Azure, este es uno basado en inteligencia artificial y se encuentra dentro del grupo de Azure Cognitive Services, su funcionalidad es interpretar, procesar y convertir audio en texto, texto en voz y realizar traducción en tiempo real [34].

4.4.4. Backend

El Backend es la capa de una aplicación que se encarga de manejar y manipular los datos, normalmente se aplican conocimientos de bases de datos, seguridad y frameworks [35]. Un framework es un marco de software que contiene código ya escrito, revisado y estandarizado, cuya función es facilitar y acelerar el desarrollo de funciones [36]. Para el desarrollo del presente TIC, se debe desarrollar un API, que es el intermediario en comunicar el entorno inmersivo con la lógica del Backend [37].

- **Tecnologías y lenguajes Backend**

Para la selección de tecnologías y lenguajes todo depende del tipo de Backend que se quiera construir [37], dentro de los lenguajes de lado del servidor tenemos a Java, C#, PHP. Node.JS, entre otros [35]. Node.JS es una plataforma de código abierto basada en JavaScript que tiene un flujo de datos asíncrono, lo que significa que una salida o petición no espera a otra, y, por ende, no almacena datos en búfer. Es altamente escalable y tiene un rendimiento destacable [38].

Django es un framework basado en Python, que al ser un lenguaje de programación de alto nivel y con una sintaxis intuitiva y bien documentada, lo convierte en una opción ideal para el desarrollo ágil de aplicaciones web [39].

Spring Boot es otro marco de trabajo, basado en Java, que permite desarrollar backend orientado a modularidad o microservicios. Se utiliza sobre todo en el campo del Internet de las Cosas y desarrollo web, donde permite integrar microservicios para crear sistemas escalables y mantenibles [40].

- **Comparativa de frameworks Backend**

Conociendo los principales frameworks de desarrollo Backend, la **Tabla 5** resume y compara estos marcos de trabajo, para facilitar su análisis en ventajas y desventajas.

Tabla 5.

Comparación de frameworks Backend

Framework	Lenguaje	Escalabilidad	Facilidad de uso	Rendimiento	Caso de uso común
Node.js [41]	JavaScript	Alta	Fácil	Alto	Aplicaciones en tiempo real.
Django [42]	Python	Moderada	Fácil	Alto	Desarrollo web.
Spring Boot [43]	Java	Alta	Moderada	Alto	Entornos empresariales

4.5. Herramientas Complementarias

Adicional a las herramientas de desarrollo mencionadas en la sección anterior, se utilizaros más softwares que ayudaron a culminar el TIC que se presentan a continuación:

4.5.1. Adobe Photoshop

Adobe Photoshop⁵ es un software de edición de imágenes, ampliamente utilizada en la comunidad de la edición y manipulación de fotos. En este proyecto, este software fue empleado para editar, preparar y recortar fotografías que fueron integradas de forma foto realista dentro del entorno virtual inmersivo desarrollado en Unity. Las tareas principales realizadas en Photoshop fue el recorte, rotación y ajuste de las imágenes, y correcciones de color y luz para evitar problemas al texturizarlas como Sprites 2D dentro de Unity.

4.5.2. Git

Git⁶ es un sistema para control de versiones distribuido que facilita la gestión de cambios en código de manera eficiente, para el desarrollo de este proyecto, git permitió mantener un seguimiento continuo de los cambios realizados en el código de Unity, por parte de los dos desarrolladores. Además, facilito la colaboración para trabajar de forma simultánea en el proyecto sin conflictos de fusión de cambios. Adicionalmente, permitió mantener un historial de cambios del proyecto, permitiendo revisar los avances realizados en cada proceso de desarrollo.

⁵ Adobe Photoshop: <https://www.adobe.com> › photoshop

⁶ Git: <https://git-scm.com>

4.5.3. GitHub

GitHub⁷ es una plataforma basada en Git, que facilita mucho el desarrollo colaborativo en proyectos de más de un desarrollador, en este caso, GitHub se encargó de almacenar el código fuente del proyecto en un repositorio de código, esto permite compartir el código y copiarlo en caso de necesitarlo. Además, gracias a la gestión de Issues o problemas, permitió organizar el trabajo de los dos desarrolladores de forma eficiente y ordenada en tres ramas distintas (la principal, y dos ramas para cada desarrollador). También, gracias a las herramientas de revisión de código como pull request, se utilizó para discusión y mejora de código antes de integrarlo en la rama principal.

4.5.4. Git Large File Storage (LFS)

Git LFS o Large File Storage⁸ es una extensión de Git que fue desarrollada con el objetivo de almacenar y manejar archivos con un tamaño que exceden los límites GitHub. Esta extensión permitió almacenar los Assets más pesados que incluyen el modelo 3D final del edificio, texturas de alta resolución, etc. Gracias a que cuenta con almacenamiento externo, redujo el tamaño del repositorio principal, lo que facilita clonar y subir cambios al proyecto.

4.6. Trabajos Relacionados

Además del documento precedente de este Trabajo de Integración Curricular, la principal fuente de información son trabajos que estén relacionados a la sensación de presencia en ambientes virtuales inmersivos y no inmersivos. La **Tabla 6** resume los Trabajos Relacionados al presente TIC.

Tabla 6.
Resumen de trabajos relacionados

Título del trabajo	Resumen
Realidad Virtual aplicada a la información del proceso académico estudiantil de la Carrera de Computación de la Facultad de Energía, las Industrias, y los Recursos Naturales no Renovables de la Universidad Nacional de Loja [1].	El trabajo base principal plantea un entorno virtual para la carrera Computación de la UNL, donde se utilizan técnicas de modelado 3D y realidad virtual. El aporte principal del trabajo, es su aporte en la medición de sensación de presencia en un entorno virtual no inmersivo, que será la base para la comparación frente a su actualización a un entorno inmersivo.

⁷ GitHub: <https://github.com>

⁸ Git LFS: <https://git-lfs.com>

Recorrido virtual de las áreas pertenecientes al campus de la Universidad Nacional de Loja [30].

El segundo aporte plantea un recorrido virtual no inmersivo del campus de la UNL, su aporte está basado en la innovación de presentar varios espacios físicos de la Universidad en un solo recorrido virtual. Se han utilizado técnicas de modelado 3D de objetos, a pesar de su poca interactividad, nos brinda una visualización de los edificios y ambientes principales de la UNL.

Visita Virtual de edificaciones arquitectónicas utilizando Realidad Virtual y Realidad Aumentada [31].

El proyecto que se ha seleccionado como tercer aporte se centra en presentar modelos creados mediante tecnologías de realidad virtual (VR) y realidad aumentada (AR). Se desarrollo una aplicación móvil para Android que permite visualizar la escuela TIC utilizando dos tecnologías. Este proyecto parte de la creación de modelos 3D de la universidad y su integración con complementos diseñados para brindar una experiencia fiel a las edificaciones reales de la institución.

Diseño e Implementación de un Serious Games con técnicas de Inteligencia Artificial para el diseño de un curso interactivo 3d de introducción a Unity [32].

El último trabajo relacionado presenta el Serious Game "Unity City" y está diseñado para enseñar a los estudiantes a crear juegos en Unity 3D utilizando tecnología de inteligencia artificial (IA). El juego fue implementado en el Centro de Capacitación y de Interfaces y Contenidos Multimedia de la Escuela de Escritura y Electrónica ESPOCH. Utilizando el método SUM.

5. Metodología

La presente sección proporciona el proceso, así como el uso de recursos para la realización y culminación del TIC. La división está descrita en cuatro secciones fundamentales, en la **sección 5.1** se detalla cuál fue el área de estudio donde se desarrolló e implementó el proyecto, por otro lado, la **sección 5.2** explica el contexto donde se desarrolló el TIC. La **sección 5.3** explica a detalle cuál fue el proceso realizado para poder cumplir con los dos objetivos principales establecidos, así como las tareas realizadas durante su desarrollo.

5.1. Área de estudio

El escenario donde se desarrolló el presente trabajo de integración curricular se ubica en el edificio de secretaría de la carrera de Computación de la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables (en adelante FEIRNNR) de la UNL, como se observa en la **Figura 3**, ubicada en las coordenadas -4.032016151476167, -79.1998889575976 de la ciudad de Loja, Ecuador, donde se tomaron las bases para la construcción del entorno virtual inmersivo.

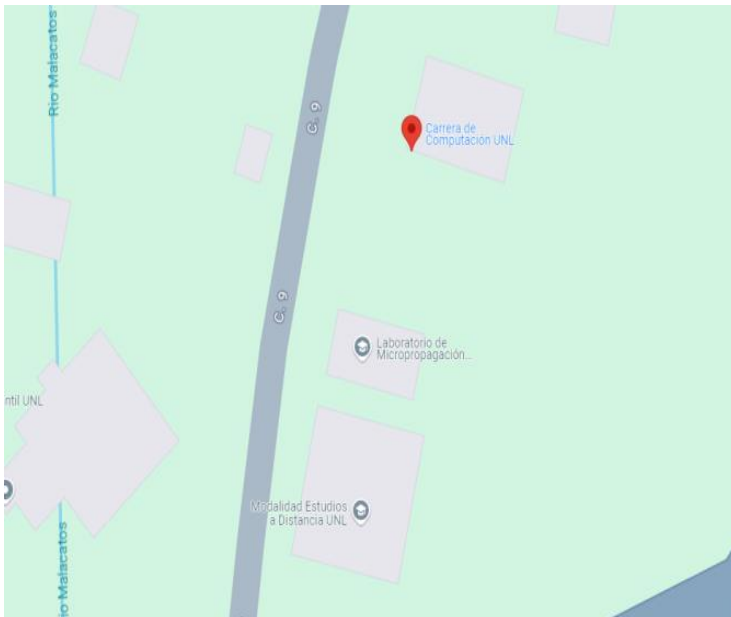


Figura 3.
Mapa de la secretaría de la carrera Computación de la UNL

5.2. Contexto

El entorno virtual inmersivo ofrece una experiencia más envolvente que el entorno no inmersivo, pues permite al usuario interactuar de manera más natural con el entorno simulado, gracias a tecnologías como gafas de realidad virtual y controles de muñeca. El presente trabajo busca actualizar el entorno con el que cuenta la carrera actualmente para responder

a la siguiente pregunta de investigación: **¿En qué medida la mejora del sistema de tour virtual no inmersivo, de la carrera de Computación de la Facultad de la Energía las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja, aumenta la sensación de presencia?**

5.3. Procedimiento

A continuación, se describe el proceso realizado para la ejecución de los dos objetivos específicos del TIC, cuya culminación contemplan el desarrollo final del objetivo general: "Actualizar el sistema de tour virtual de la carrera de Computación de la FEIRNNR a una experiencia inmersiva, mediante la implementación de realidad virtual junto con tecnologías inmersivas para aumentar la sensación de presencia dentro del entorno".

5.3.1. Objetivo específico 1: Actualizar el sistema de tour virtual de la carrera Computación de la FEIRNNR a un entorno inmersivo mediante la metodología MSD-VR.

5.3.1.1. Fase de asignación

En la fase de asignación, y siguiendo la metodología MSD-VR [33], se realizó la elicitación de requisitos funcionales, mediante las entrevistas al docente tutor Andrés Navas Castellanos junto con el director de la carrera de Computación y cotutor Pablo Fernando Ordoñez Ordoñez (véase sección *Anexo 2*). Estas entrevistas permitieron definir las necesidades a cubrir con el desarrollo del presente TIC, así como el público objetivo y los escenarios a recrear en el nuevo entorno inmersivo. También, se designaron los escenarios y tareas a recrear dentro del nuevo entorno virtual inmersivo. Finalmente, se definió un nivel de inmersividad objetivo (mayor que 6.01 en la escala de Likert con el test SUS de sensación de presencia percibida) basándonos en el trabajo preliminar de Pucha [1], junto con la elección del hardware de realidad virtual (Meta Quest 2 [34]). Además, mediante un análisis teórico comparativo (Sección 4.3), esta fase permitió la selección de tecnologías a utilizar para el desarrollo del entorno virtual.

5.3.1.2. Fase de análisis

Durante la segunda fase, se definieron las actividades y acciones que los usuarios podrán realizar dentro del entorno virtual, con la ayuda del docente tutor. Para asegurar la coherencia del sistema, se desarrollaron diagramas de clases y de componentes, además, se realizó la medida manual mediante una visita física de las dimensiones del edificio, para esto, se midió en pies para transformar las medidas a metros y poder realizar un plano detallado del edificio de secretaria.

5.3.1.3. Fase de construcción

Como tercera fase, se definió y se realizó la construcción de los activos que compondrían el entorno virtual inmersivo, con la ayuda de software de desarrollo Blender para el modelado del edificio y elementos exteriores, y el uso de Unity para la creación del entorno virtual inmersivo, y el desarrollo de los scripts necesarios. A la par, la documentación de desarrollo se muestra en la Sección 6.

5.3.1.4. Fase de implementación

Durante la cuarta fase se implementó el entorno virtual desarrollado en la fase anterior con el hardware definido en la primera fase (Meta Quest 2), que comprende el uso de las gafas inmersivas, así como los controles de muñeca que permiten la simulación de respuestas físicas acorde a los movimientos de las manos. El proceso técnico para exportar y desplegar el entorno inmersivo dentro de los Meta Quest 2 se describe a continuación.

- **Preparación de entorno de desarrollo**

Una vez terminada la construcción de proyecto en Unity, es necesario habilitar el modo desarrollador de los lentes de realidad aumentada. Este modo, permite instalar aplicaciones externas desarrolladas para esta plataforma. La activación de este modo está sujeta al uso de aplicaciones externas desarrolladas por Meta, la **Tabla 7** describe el nombre y uso de estas aplicaciones.

Tabla 7.

Resumen de uso de aplicaciones de Meta para el desarrollo de aplicaciones para Meta Quest 2

Aplicación	Plataforma	Uso
Meta Horizon	Móvil	Se utiliza para configurar los Meta Quest 2 y activar el modo desarrollador, es necesario usar la misma cuenta de Meta Developer propietaria de los lentes.
Meta Quest Link	PC	Esta herramienta es esencial para conectar los Meta Quest 2 a la computadora, permite el reconocimiento de este hardware para realizar pruebas en Unity y acceder a información básica como estado de carga del dispositivo y controles, calidad de conexión, etc.
Meta Developer Hub	PC	Esta plataforma es un centro de control exclusivo para desarrolladores, desde aquí, se pueden agregar y gestionar dispositivos, ver los logs de salida, monitorear el rendimiento y aplicar configuraciones avanzadas para exportar el APK.

La **Figura 4** y **Figura 5** muestran la evidencia de la conexión de los Meta Quest 2 a las aplicaciones de Meta necesarias para el desarrollo de aplicaciones de realidad virtual inmersiva.

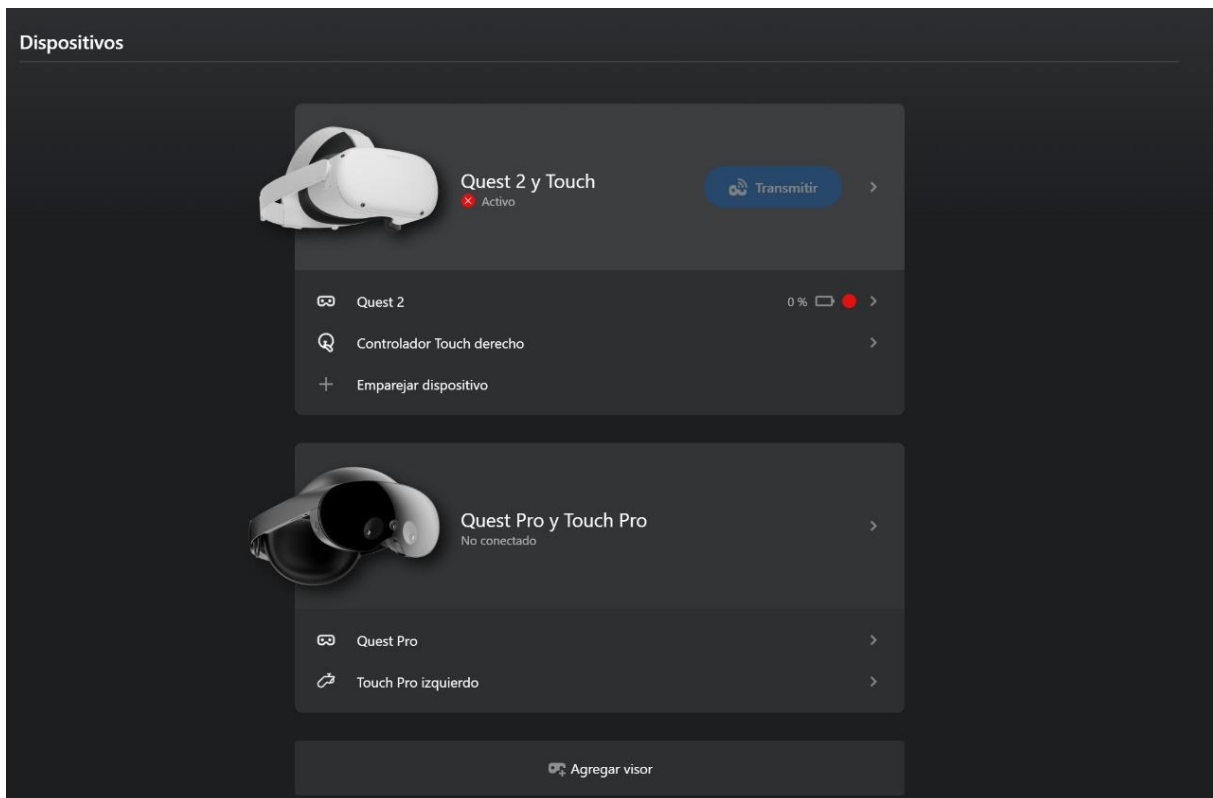


Figura 4.
Dispositivos Meta Quest 2 conectados exitosamente a Meta Quest Link

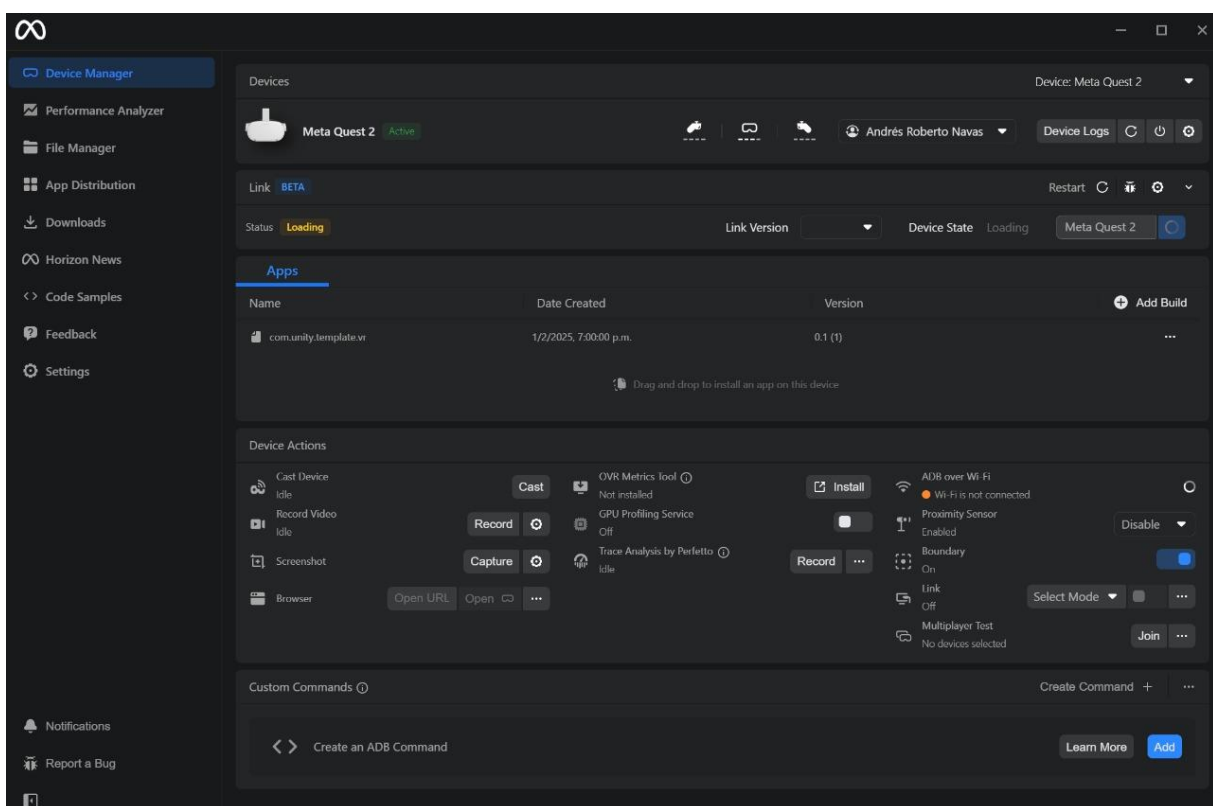


Figura 5.
Conexión exitosa de Meta Quest 2 a Meta Developer Hub

- Configuración en Unity

Cuando los lentes se encuentran en modo desarrollador, se debe configurar Unity para exportar el entorno inmersivo, para esto, se debe instalar el SDK de Oculus mediante el administrador de paquetes, así como el SDK de Android para Unity que permite exportar en formato APK. Luego, se debe seleccionar la opción de "Build Settings", dentro de Unity, con los lentes conectados, la **Figura 6** muestra la configuración utilizada para exportar el proyecto, así como la selección del dispositivo Meta Quest 2 (Oculus Quest 2).

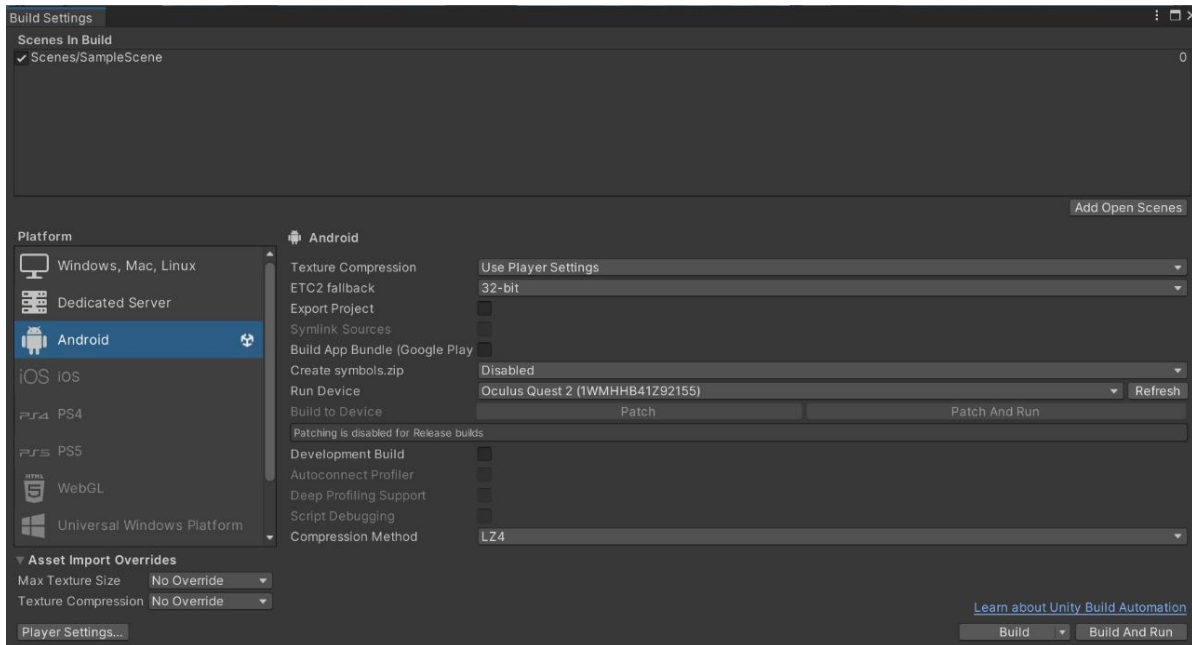


Figura 6.
Configuración de Unity para exportar el entorno inmersivo

5.3.1.5. Fase de pruebas

En esta fase se aplicaron pruebas de usuario, con el objetivo de recibir las recomendaciones y comentarios pertinentes respecto a cómo las personas se sienten al probar el software. Con este fin, se aprovechó la participación en el evento "VR Day Latam" realizado en la Universidad Nacional del Chimborazo para presentar el software a posibles usuarios finales, a pesar de tener el proyecto en fase de desarrollo, esto permitió recabar información y sugerencias de mejora. La presentación, realizada en un stand en representación de la Universidad Nacional de Loja, a lo largo del evento permitió evaluar el software con diez personas, con las cuales se realizó el siguiente procedimiento:

- Los lentes de realidad virtual Meta Quest 2 fueron colocados por los participantes.
- Con ayuda de una integrante del stand, se ajustó el dispositivo a las medidas y distancias más prácticas para el usuario

- Las acciones disponibles, como caminar y hablar con los NPC's, fueron explicadas a los usuarios.
- Los usuarios procedieron a recorrer el entorno virtual y a interactuar con los diferentes NPCs que se encontraban a lo largo del terreno del juego
- Tras pasar 10 minutos, se les retiró el dispositivo y se procedió a anotar el feedback, retroalimentación o comentarios que proporcionaron los usuarios tras utilizar el software, los más relevantes se recopilaron y se presentan en la **Tabla 13** ubicada en la sección de resultados.

En la sección **Anexo 3** se observa como evidencia a uno de los participantes de la conferencia utilizando los lentes de realidad virtual, mientras interactúa con la aplicación desarrollada. El participante está equipado con los controles del dispositivo, con los cuales pudo desplazarse por el entorno virtual e interactuar con los NPCs del juego. Además, en el stand se encuentra el equipo de desarrollo asistiendo al usuario en caso de cualquier duda o conflicto que surgiera mientras el este probaba el sistema.

5.3.2. Objetivo específico 2: Implementar un sistema de respuestas inteligentes y proactivas, con el modelo generativo "Gemini 1.5 Pro", para los avatares informativos dentro del sistema virtual inmersiva.

5.3.2.1. Fase de asignación

En esta fase inicial, se definieron los elementos fundamentales para el desarrollo del sistema de respuestas inteligentes. El proceso comenzó con reuniones entre el equipo de desarrollo y los tutores académicos con el propósito de identificar las necesidades específicas del sistema. Se determinó que el flujo de trabajo debería incluir varios pasos clave: capturar la voz del usuario mediante un micrófono integrado, convertir este audio en texto para poder ser interpretado por el modelo de inteligencia artificial, procesar el texto con el modelo generativo "Gemini 1.5 Pro" interactuando con él a través de un sistema backend que facilite la comunicación entre el entorno virtual y el servicio de inteligencia artificial, para finalmente convertir la respuesta generada en un archivo de audio que se reproducirá dentro del entorno virtual. La selección de tecnologías fue un punto crítico en esta fase, para esto se realizó un análisis comparativo entre las diferentes tecnologías u ofertas del mercado de desarrollo de software, detallando los resultados obtenidos en la sección 4.4.4.

Tras realizar este análisis se logró determinar cuál es la mejor alternativa para poder implementar el sistema backend, de igual forma se logró seleccionar que sistemas o servicios serán los que faciliten el proceso de conversión y manipulación tanto de texto a audio como

viceversa, todas las herramientas seleccionadas se pueden ver en la sección de resultados 6.2.

5.3.2.2. Fase de análisis

Durante esta fase, se llevó a cabo un análisis exhaustivo del flujo de trabajo requerido para garantizar una interacción eficiente entre el usuario y el sistema. El proceso se estructuró en 7 etapas principales:

- Detección de la voz del usuario mediante un micrófono integrado en el dispositivo de realidad virtual.
- Conversión del audio capturado a texto utilizando la API de Hugging Face, asegurando la precisión en la interpretación del lenguaje natural.
- Envío del texto obtenido hacia el API del backend a través de una consulta HTTP de tipo PUT
- Procesamiento del texto generado en el backend desarrollado en Node.js, que se conecta a la API de "Gemini 1.5 Pro" para generar respuestas coherentes y contextualizadas.
- El backend devuelve la respuesta generada por la inteligencia artificial en formato de texto a través de una respuesta de tipo JSON
- El sistema del entorno virtual recibe el texto generado por el api y se procede a la conversión del texto a un archivo de audio mediante Azure Speech, utilizando configuraciones personalizadas para garantizar una voz natural y acorde al avatar.
- Reproducción del audio dentro del entorno virtual, sincronizando las animaciones de los modelos NPC'S con las respuestas generadas

Se diseñaron diagramas de actividades para modelar estas interacciones, lo que permitió identificar posibles cuellos de botella y optimizar cada etapa. Además, se definieron las interfaces entre los componentes del sistema, asegurando que las APIs y el backend se integraran de manera eficiente con Unity.

5.3.2.3. Fase de construcción

La construcción del sistema comenzó con la implementación del backend utilizando Node.js y Express. Se configuró el endpoint necesario para manejar las solicitudes provenientes del entorno virtual. Esta ruta consistió en uno que procesara el texto recibido enviado tras ser generado con el api de Hugging Faces y enviándolo a "Gemini 1.5 Pro", para que este modelo se encargue de procesar la respuesta necesaria y la retorne a través de un formato de archivo JSON mediante una respuesta HTTP.

En paralelo, se desarrollaron scripts en Unity para gestionar la interacción entre los avatares y el backend. Para esto se hizo uso de uno de los objetos de Unity conocido como Box Collider, con el cual se puede registrar cuando un jugador ha entrado dentro de este y por lo tanto poder ejecutar un script. A partir de estos se asignó la lógica en la que al momento de que un jugador este dentro del collider de un personaje no jugable o por sus siglas N.P.C., se va a estar escuchando constantemente el evento de que se presione el botón para hablar, que en este caso es el botón B del controlador derecho. La **Figura 7** muestra el botón mencionado, de igual forma, para detectar el input del controlador se partió del sistema de detección de entradas de unity y el paquete de VR instalado con el cual la detección y clasificación de los botones ya estaba integrada, esto permitió que el manejo de cuando el usuario desea hablar y cuando el sistema se lo permite sea mucho más intuitivo y fluido.



Figura 7.
Control derecho de dispositivo Meta Quest 2 [44]

5.3.2.4. Fase de implementación

En esta fase, se integraron todos los componentes desarrollados en un sistema cohesivo. El backend fue desplegado en un servidor en la nube, garantizando su disponibilidad y escalabilidad. Dentro de Unity, se realizaron ajustes finales para optimizar la sincronización entre las respuestas generadas y las animaciones de los avatares. Se calibraron los parámetros de Azure Speech para mejorar la calidad del audio y garantizar que las voces utilizadas fueran consistentes con la personalidad de cada avatar.

5.3.2.5. Fase de pruebas

Para esta fase se utilizó el cuestionario SUS (System Usability Scale) como herramienta principal para evaluar la percepción de usabilidad e inmersión del sistema.

Primero, se definió el tamaño de la muestra a la que se aplicaría el test, para esto se aplicó la fórmula de población finita, que se estableció en el número aproximado de estudiantes que puedan mostrar interés en la facultad seleccionados a conveniencia como se observa en (1).

$$n = \frac{N\sigma^2 Z^2}{(N-1)e^2 + \sigma^2 Z^2} \quad (1)$$

- **Datos de muestreo**

Tamaño del universo (N) = 520

- **Fórmula de muestra (n)**

Nivel de confianza (Z) = 1.95

Nivel de error (e) = 0.15

Desviación (σ) = 0.5

Entonces, definiendo un aproximado de 520 estudiantes interesados en conocer información sobre la Facultad FEIRNNR tenemos una muestra de:

$$n = \frac{520 * (0.5)^2 * (1.95)^2}{(520 - 1)(0.15)^2 + (0.5)^2 * (1.95)^2}$$
$$n = \frac{494.33}{12.63}$$
$$n = 39.14$$

Aproximado a 39 estudiantes para realizar las encuestas. Conociendo el tamaño de la muestra, se definió un procedimiento detallado para aplicar las pruebas, que incluyó los siguientes pasos:

1. Selección de un grupo de usuarios representativo de estudiantes de la carrera de Computación.
2. Explicación previa sobre el funcionamiento del sistema, destacando cómo interactuar con los avatares y las funcionalidades disponibles.
3. Los usuarios realizaron tareas específicas en el entorno inmersivo, como hacer preguntas a los avatares y evaluar la naturalidad de las respuestas generadas.
4. Aplicación del cuestionario SUS, compuesto por 6 preguntas que miden las respuestas relacionadas a la inmersión y SoP en una escala del 1 al 7.
5. Cálculo de Alfa de Cronbrach para medir la fiabilidad de los resultados de las encuestas y promedio final de SoP.

El cuestionario permitió recopilar datos cuantitativos sobre la SoP percibida. Estos datos fueron recopilados y tabulados, con el objetivo de obtener un promedio de SoP y compararlo con el resultado del entorno no inmersivo anterior. Además, durante la aplicación del cuestionario, se tomaron notas sobre los comentarios cualitativos proporcionados por los usuarios, lo que ayudó a entender de manera más profunda las necesidades y expectativas respecto al sistema.

Los resultados de estas pruebas permitieron identificar el impacto de migrar un entorno no inmersivo a uno inmersivo sobre todo en la SoP. Finalmente, se documentaron todos los hallazgos y se realizaron iteraciones sobre el sistema para abordar cualquier problema identificado durante las pruebas. Esta fase aseguró que el sistema estuviera completamente preparado para su despliegue en el entorno virtual inmersivo.

6. Resultados

A continuación, se presentan los resultados obtenidos al ejecutar los objetivos definidos mediante la metodología establecida:

6.1. Objetivo 1: Actualizar el sistema de tour virtual de la carrera Computación de la FEIRNNR a un entorno inmersivo mediante la metodología MSD-VR.

6.1.1. Fase de asignación

- **Elicitación de requisitos**

A partir de las entrevistas realizadas a los docentes Andrés Roberto Navas Castellanos y Pablo Fernando Ordoñez Ordoñez el día miércoles 10 de junio del 2024, se redactaron los requisitos correspondientes. La **Tabla 8** muestra los requisitos funcionales y la **Tabla 9** establece los requisitos no funcionales a cubrir por el sistema inmersivo. Además, se definió como público objetivo a todos los usuarios (sean docentes, estudiantes, visitantes, etc.) que tengan interés en interactuar con el entorno inmersivo. El sistema virtual inmersivo deberá permitir al usuario:

Tabla 8.
Requisitos funcionales del tour virtual

Código	Requisitos funcionales	Descripción
RF001	Acceder a un recorrido virtual por el bloque administrativo.	El tour virtual será libre dentro del límite del edificio administrativo de la facultad de energía.
RF002	Visualizar el espacio del departamento.	Los usuarios podrán explorar y visualizar las salas, espacios y oficinas correspondientes al edificio de secretaría de la facultad.
RF003	Seleccionar los distintos puntos interactivos.	Mediante una interacción con los controles de Meta Quest 2 o detección de triggers.
RF004	Iniciar conversaciones interactivas con los NPC de acuerdo a la sección que corresponda (administrativa, académica e informativa).	La conversación será iniciada mediante un botón para la detección y transcripción de voz a texto.
RF005	Visualizar recursos informativos generales de la facultad de Energía.	Los recursos, en formato textual o audiovisual, incluyen videos, información de contacto, etc.
RF006	Visualizar indicaciones sobre cómo realizar el tour virtual.	Las indicaciones, o "hints", brindan ayuda para los usuarios sobre cómo navegar, como iniciar conversaciones, etc.

Tabla 9.
Requisitos no funcionales del tour virtual

Código	Requisito no funcional	Categoría
RNF001	El sistema deberá utilizar una paleta de colores, en logos y dentro del entorno, que sean coherentes con los colores de la carrera.	Usabilidad
RNF002	El entorno virtual deberá utilizar fuentes de texto adecuadas y que sigan la estética general del ambiente.	
RNF004	El entorno virtual deberá integrar el uso de controles con dispositivos de realidad virtual externos (Meta Quest 2).	Compatibilidad
RNF005	El entorno virtual deberá integrar Hints y ToolTips informativos de una forma estética y no invasiva.	Usabilidad
RNF006	El entorno virtual deberá poseer un sistema de colisiones lo más aproximado a la realidad como sea posible.	Funcionamiento
RNF009	El entorno virtual deberá implementar texturas de calidad media-alta, para mejorar la sensación de realismo.	Funcionamiento
RNF010	El entorno virtual deberá presentar una interfaz adecuada y sin sobrecarga de objetos y elementos visuales (videos, imágenes, etc.).	Usabilidad
RNF011	El entorno virtual deberá integrar sonidos ambientales no invasivos.	
RNF012	El entorno virtual deberá delimitar las áreas no interactivas o inaccesibles del usuario mediante técnicas poco intrusivas	Funcionamiento
RNF013	El entorno virtual deberá integrar chats interactivos (con uso de modelos de IA Generativa) con los NPC.	Interactividad

- **Definición de escenarios a recrear**

En la **Tabla 10** se describen los escenarios establecidos a recrear, que son los mismos del tour virtual no inmersivo preliminar [1], sumando al nuevo escenario de sala de decanato.

Tabla 10.
Escenarios a recrear dentro del tour virtual

<i>Escenario</i>	<i>Descripción</i>
Exteriores del Edificio Administrativo	El escenario principal, que incluye aceras, jardines, caminos y demás elementos externos al edificio. Además, se incluyen las palmeras, césped, concreto y cielo.
Fachada del Edificio Principal	Incluye la fachada externa y detallada del edificio administrativo, incluyendo los diseños arquitectónicos y colores representativos del mismo.
Vestíbulo del Edificio Principal	Entrada principal, diseño de los interiores, como la recepción, elementos decorativos, sillas, señalizaciones, etc.
Sala de Secretaría de las carreras de Computación, Electromecánica, Mecánica Automotriz y Electrónica y Telecomunicaciones	Oficina principal de la secretaría de cada carrera, que incluye escritorios, sillas, archivadores, computadoras y demás decoraciones.
Auditorio o Sala de Ponencias	Sala orientada a la presentación y ponencia de trabajos de titulación, incluyendo los elementos correspondientes como sillas, proyector, decoraciones, etc.
Oficina de Decanato	Área de trabajo del Decano de la facultad.
Oficina de Coordinación Administrativa y Financiera	Sala orientada a la información de administración financiera de la facultad

- **Selección de motor gráfico**

Luego del análisis teórico (Sección 4.3.2), se optó por el uso de Unity3D, que es un motor para renderizado de entornos virtuales inmersivos 3D y 2D [45]. Unity ha destacado entre las diferentes opciones en el ámbito de la programación de aplicaciones y videojuegos debido a su gran popularidad y por lo mismo a su amplia información respecto a cómo desarrollar aplicaciones, pues cuenta con “Addons” o herramientas extras que ya proveen las bases para crear este tipo de software sin necesidad de enfocarse en los aspectos iniciales o básicos, además, posee una basta cantidad de documentación que facilitan el entendimiento de esta herramienta [46]. Por otro lado, su integración con el lenguaje de C# implementa diferentes métodos o funciones que aceleran el desarrollo, pues acciones como implementar consultas de API, para la parte de respuestas interactivas, resulta en un proceso eficiente y efectivo, que minimiza la complejidad frente a otras herramientas [20].

- **Selección de herramienta de modelado 3D**

Gracias al análisis teórico (Sección 4.3.3), la herramienta seleccionada fue Blender, que es un paquete de estudio para el modelado y animación 2D y 3D, permite generar de forma flexible y eficiente modelos complejos haciendo uso de técnicas como UV Mapping o referencia de imágenes de gran calidad [21]. Este software es de uso gratuito y de código abierto, por lo que es conocido y usado en muchas áreas del modelado 3D, lo que equivale a que exista una amplia variedad de guías y tutoriales que simplifiquen la comprensión y el uso correcto de la herramienta [22]. Además, otra de las ventajas que supone el uso de esta herramienta es por su capacidad para integrarse fácilmente con Unity3D permitiendo así una funcionalidad cruzada, lo que agiliza transferir modelos y animaciones entre ambas plataformas, proporcionando una base sólida para crear animaciones detalladas y realistas necesarias para una experiencia de realidad virtual inmersiva.[23]

6.1.2. Fase de análisis

A continuación, se muestran los resultados de la fase análisis, que corresponden al modelado de diagrama de clases, acciones que se podrán realizar, etc.:

- **Diagrama de Clases**

La **Figura 8** muestra el diagrama de clases desarrollado en Diagrams.net para el tour virtual, donde se define la estructura estática del sistema, representado mediante clases con atributos y métodos. Las relaciones entre clases sirven como una base para implementar el sistema, brindando una visión general de la lógica de funcionamiento y como se relacionarán las diferentes partes del sistema entre sí.

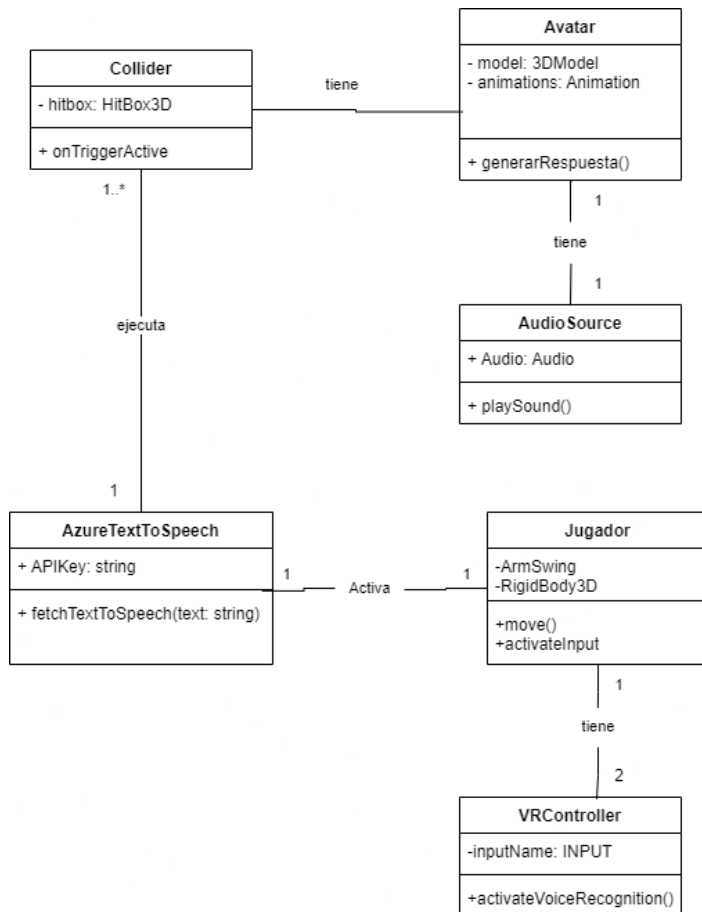


Figura 8.
Diagrama de clases del entorno virtual inmersivo

En la **Tabla 11** se describen las clases del diagrama con sus respectivos atributos y métodos.

Tabla 11.
Clases y atributos del diagrama

Clase	Atributos	Métodos
Collider	hitbox: HitBox3D	onTriggerActive()
Avatar	model: 3DModel animations: Animation	generarRespuesta()
AudioSource	Audio: Audio	playSound()
AzureTextToSpeech	APIKey: string	fetchTextToSpeech(text: string)
Jugador	ArmSwing Rigidbody3D	move() activatelnput()
VRController	inputName: INPUT	activateVoiceRecognition()

- **Diagrama de componentes**

Con la ayuda de la herramienta Diagrams.net, la **Figura 9** muestra el diagrama de componentes elaborado para el entorno virtual inmersivo

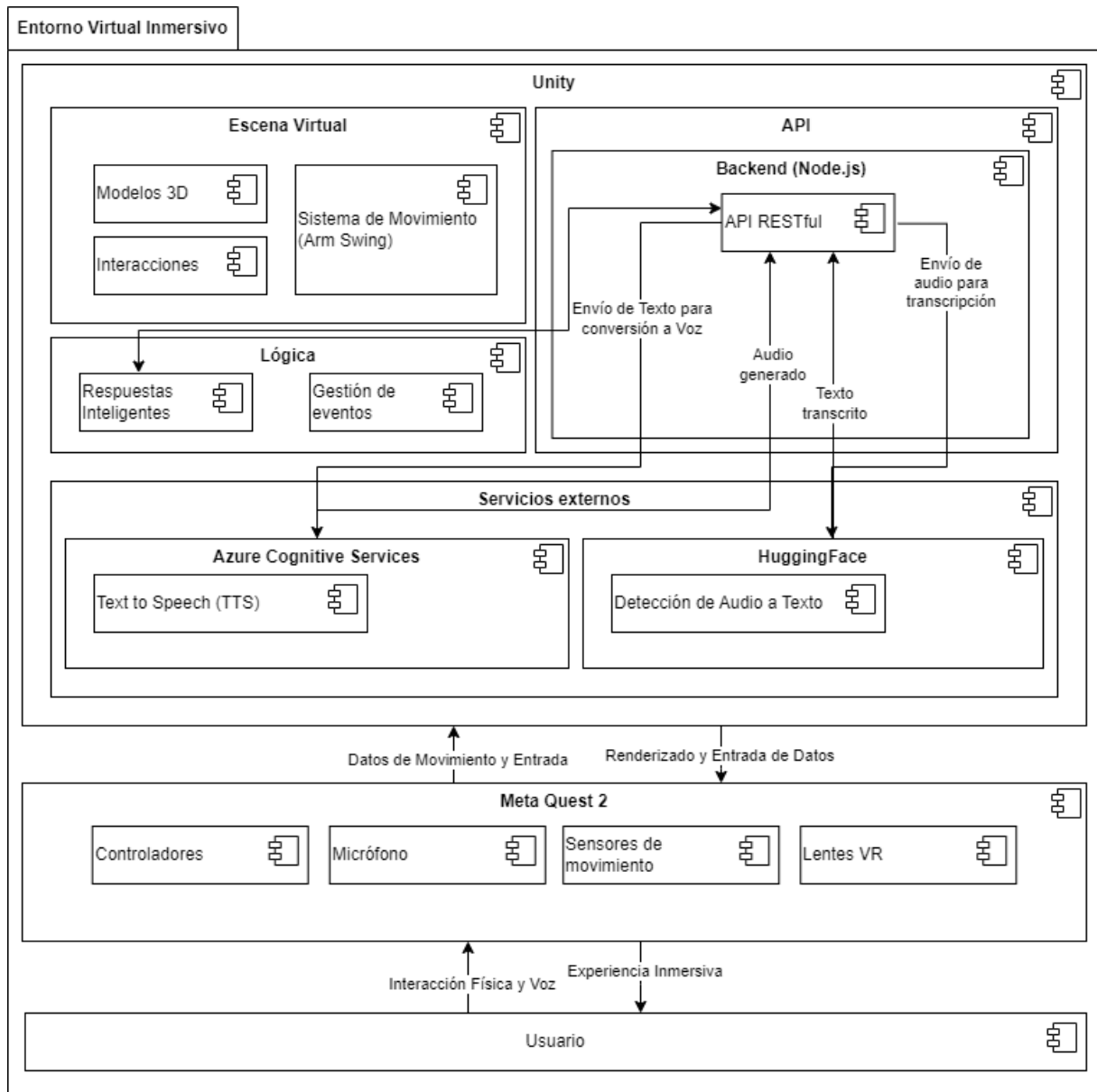


Figura 9.
Diagrama de componentes de entorno virtual inmersivo (Elaboración propia)

- **Planos del edificio**

Luego de la toma de medidas manuales se procedió a diagramar en la herramienta LucidChart los planos del edificio. La **Figura 10** presenta un plano detallado del edificio, de esta forma, se estructuraron las acciones y objetos necesarios para crear una experiencia inmersiva y funcional.

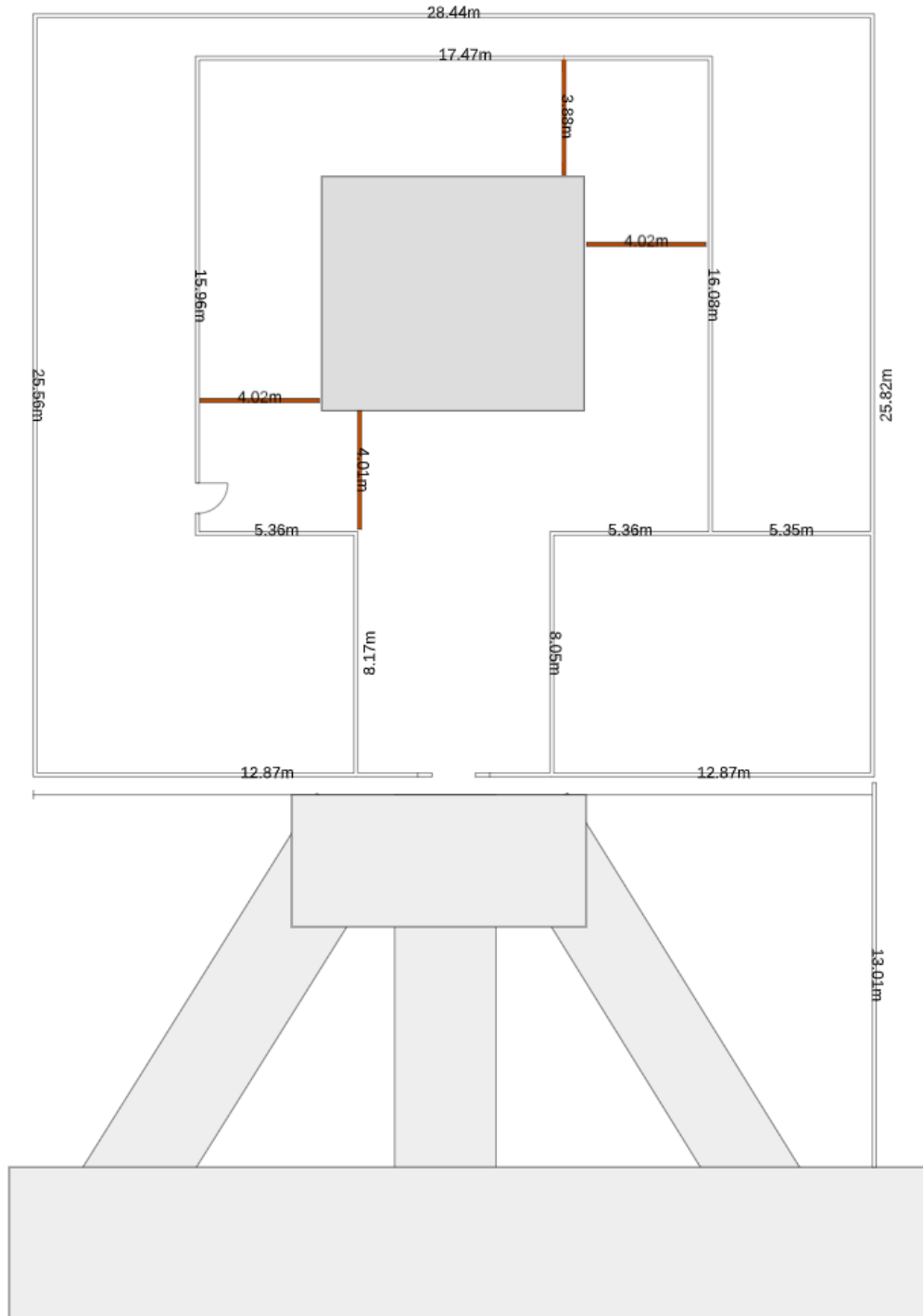


Figura 10.
Planos del edificio de secretaría de la facultad (Elaboración propia)

6.1.3. Fase de construcción

- **Modelado principal en Blender**

Mediante la implementación de la herramienta de Blender y haciendo uso de los planos estimados (**Figura 10**), se modeló el edificio de secretaría de la FEIRNNIR, la **Figura 11** presenta la ruta de archivos del proyecto de Blender, estructurado en cuatro partes principales

que permitieron llevar una mejor gestión y control del modelo, tanto a la hora de exportarlo al motor gráfico como al momento de asignarle sus texturas, dimensiones y demás factores. Además, también facilitó el proceso del modelado en sí, pues gracias a la posibilidad de activar y desactivar la visibilidad de estas capas, fue más accesible y sencillo enfocarse en cada una por separado, a continuación, se describe cada una de estas capas.

- **Paredes exteriores:** En esta sección, se encuentran los modelos correspondientes a las paredes externas del edificio. Estas estructuras incluyen tanto las dimensiones como las texturas aplicadas para representar de manera realista el aspecto exterior de la secretaría. Los elementos fueron organizados con base en los planos proporcionados, considerando detalles de color, textura y acabados que se asemejan a la construcción real.
- **Interior:** Este apartado contiene los elementos que configuran el diseño interior, tales como divisiones internas, puertas y ventanas. Cada objeto fue modelado para representar fielmente la disposición y características de los espacios dentro del edificio. Los materiales y texturas se aplicaron para que los usuarios puedan visualizar una experiencia cercana a la realidad, destacando detalles específicos del diseño arquitectónico.
- **Piso:** En la carpeta correspondiente al piso, se encuentran los modelos que replican la superficie de cerámica utilizada en el edificio. Se aplicó una textura cerámica que reproduce tanto el color como el patrón de las baldosas, alineándose con los detalles específicos del acabado real. La disposición de las baldosas fue diseñada para reflejar la estructura visual y la disposición observada en los planos arquitectónicos, permitiendo una representación precisa y detallada del piso dentro del modelo.
- **Techo:** Finalmente, en esta sección se encuentran los elementos que conforman la estructura del techo, incluidos los detalles de diseño y materiales. Este componente se modeló siguiendo las especificaciones de altura y pendiente según los planos, proporcionando una representación completa del edificio.

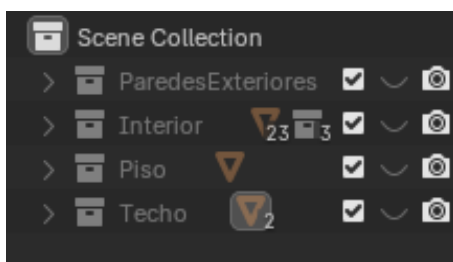


Figura 11.
Ruta de archivos de proyecto del edificio en Blender

Luego de completar el modelado estructurado en Blender y dividir el edificio en sus cuatro componentes principales, se procede a describir cada uno de ellos en detalle, utilizando imágenes que ilustran visualmente el resultado de este proceso. Cada Figura representa un aspecto específico del edificio modelado y proporciona una perspectiva precisa de los elementos que conforman la estructura arquitectónica. A continuación, se presenta un análisis de cada parte, destacando sus características, el diseño y las decisiones de modelado adoptadas en cada caso para garantizar una representación realista y coherente con los planos arquitectónicos iniciales.

La **Figura 12** presenta el modelado 3D de las paredes exteriores del edificio de secretaría. Estas paredes fueron diseñadas considerando la estructura arquitectónica del edificio, incluyendo detalles como las ventanas con arcos y el marco en azul que les otorga un contraste distintivo lo más parecido al edificio ubicado en la universidad. Los muros presentan una textura y un color que imitan el acabado real, logrando una apariencia cercana a la realidad. Las ventanas han sido modeladas con rejas, agregando elementos de seguridad y realismo que reflejan las características físicas del edificio. La entrada principal se encuentra en el centro de la fachada, resaltada con una estructura de escalones, que aporta profundidad y funcionalidad al modelo. Cada uno de estos elementos fue cuidadosamente modelado para representar fielmente las proporciones y detalles observados en los planos arquitectónicos, logrando una base sólida para el resto del diseño estructural.

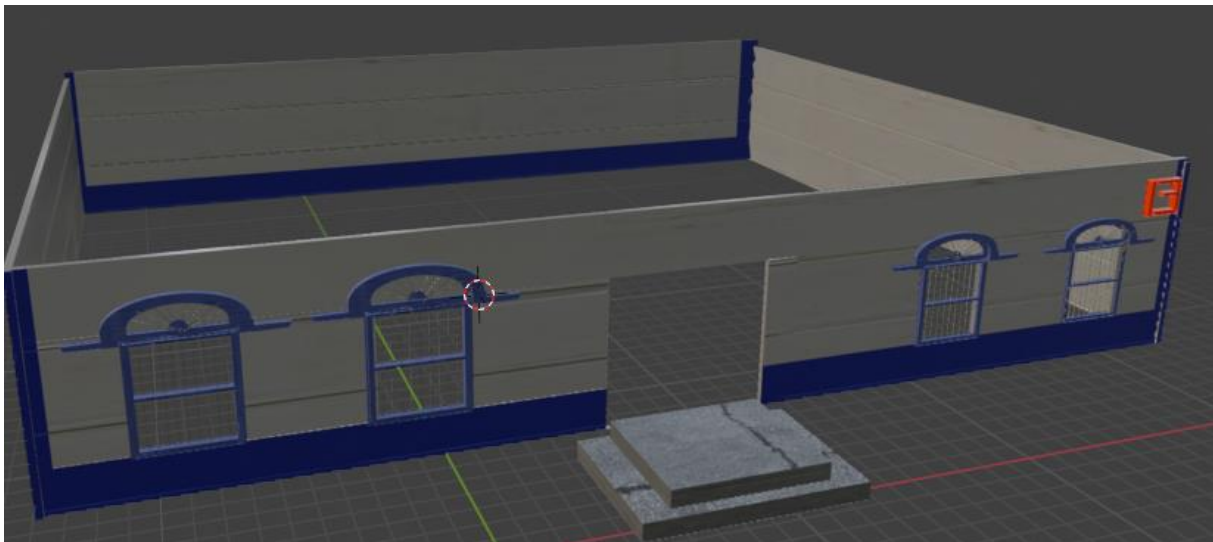


Figura 12.
Modelado de paredes exteriores del edificio de secretaría

La **Figura 13** muestra el modelado interior del edificio de secretaría, en el cual se aprecian las divisiones internas y detalles arquitectónicos característicos de los espacios interiores. Las paredes internas fueron modeladas con una textura y color que simulan los acabados de concreto, logrando una representación realista de la estructura. Se incluyen

además columnas y pilares, los cuales son elementos estructurales que otorgan estabilidad y soporte al modelo, replicando las características de la construcción real. Asimismo, las puertas fueron diseñadas para ajustarse a la disposición interna, brindando acceso y separación entre las áreas. Estos detalles permitieron presentar un trabajo con un nivel de precisión alto en el modelado, respetando tanto la disposición espacial como los materiales presentes en el diseño original, logrando una representación que facilita la visualización de los espacios interiores del edificio.

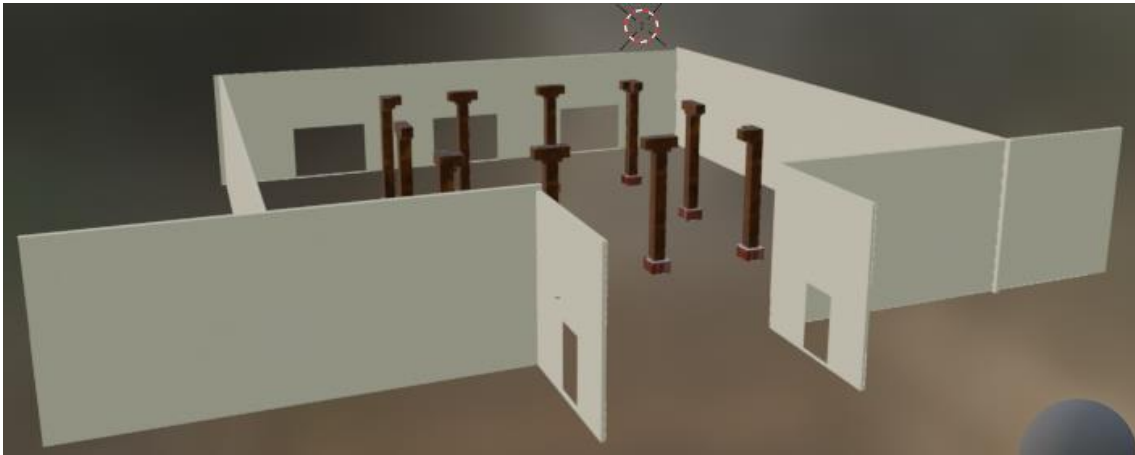


Figura 13.
Modelado de paredes interiores del edificio de secretaría

En la **Figura 14** se muestra el modelado del piso del edificio, donde se evidencia como fue diseñado con una textura de baldosas de cerámica que reproduce fielmente el acabado del suelo real, utilizando dos texturas diferentes para representar los bordes de las paredes internas del edificio, logrando así un nivel de detalle más elevado y en general proporcionando un diseño más complejo e interesante al modelo en general. Las baldosas presentan un patrón en tonos marrones y beige en sentido diagonal, lo que aporta un aspecto cálido y tradicional al espacio, alineado con los materiales utilizados en la construcción física. En el centro, se observa una sección diferenciada que corresponde a una zona central la cual tienen en su mayoría un acabado con la baldosa de tono marrón y conecta con el tragaluz del techo. Este diseño se llevó a cabo siguiendo el patrón y la disposición especificados en los planos, logrando una representación detallada del suelo. La elección de una textura cerámica no solo mejora el realismo del modelo, sino que también permite una visualización cercana al material verdadero, ofreciendo una experiencia más inmersiva.

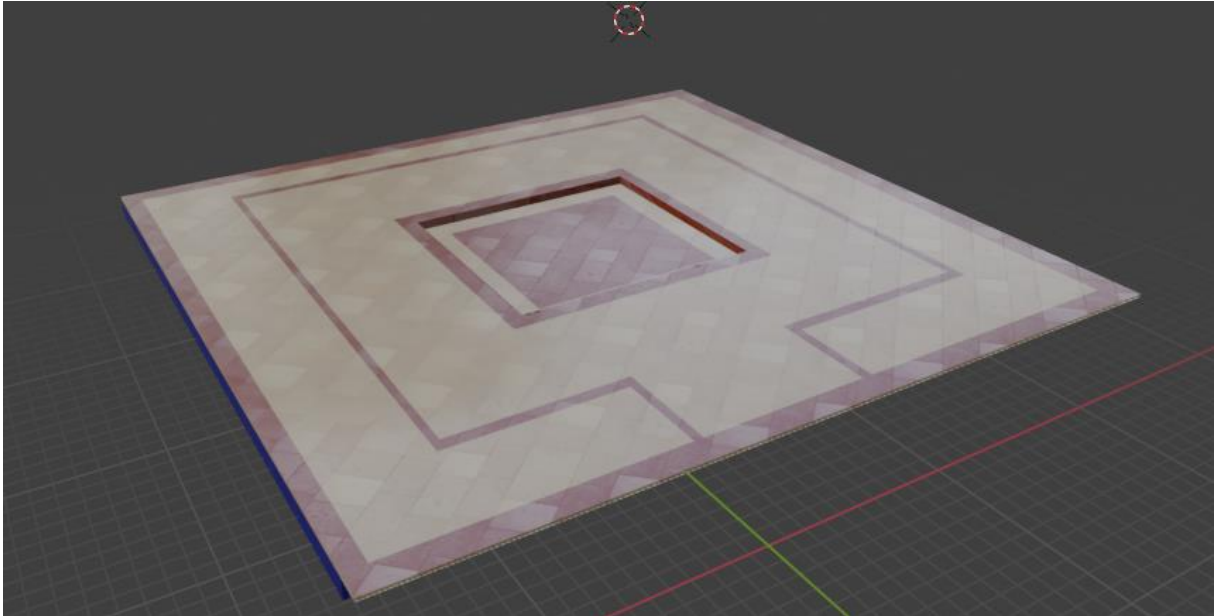


Figura 14.
Modelado del piso del edificio de secretaría

Para el techo, en la **Figura 15** se observa el modelo estructural que sigue las especificaciones arquitectónicas del edificio de la secretaría. El diseño incluye una pendiente hacia el centro, en donde se ha aplicado una textura de teja marrón para simular el acabado externo. Esta textura proporciona un aspecto rústico y realista, alineándose con los materiales empleados en la construcción física del techo. En el área central, se ha modelado una estructura de superficie reflectante que representa el tragaluz del edificio, permitiendo la entrada de luz natural al interior. Este componente añade profundidad al diseño y asegura que el techo cumpla con las especificaciones de los planos arquitectónicos, tanto en términos de inclinación como de materiales. Cada detalle fue cuidadosamente seleccionado para reproducir la funcionalidad y estética del techo real, asegurando una experiencia inmersiva y realista dentro del entorno virtual.

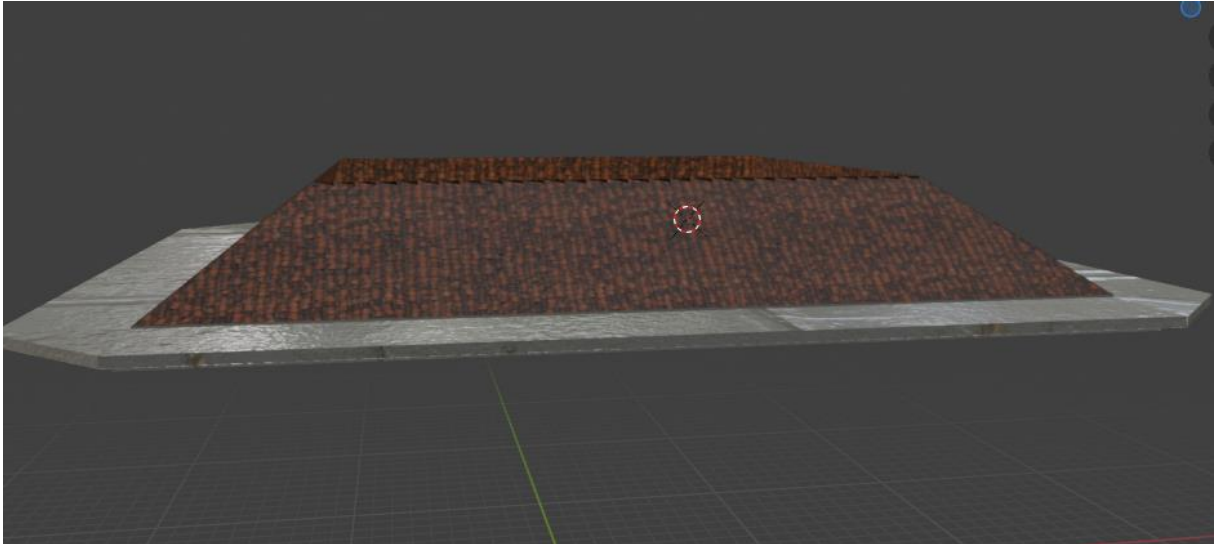


Figura 15.
Modelado del techo del edificio de secretaría

Finalmente, en la **Figura 16** se observa el modelo final del edificio de la secretaría de la facultad, integrado completamente en el entorno 3D de Blender. Este modelo incluye tanto los elementos exteriores como los interiores, siguiendo un diseño modular que optimiza su incorporación en un entorno inmersivo. En cuanto a las paredes exteriores, se aprecia una estructura detallada con ventanas arqueadas que incluyen marcos en azul, un contraste que aporta identidad visual al edificio sin sobrecargar el procesamiento gráfico en el entorno VR.

Este diseño no solo es estéticamente acorde a la estructura original, sino que también aporta iluminación al interior en el entorno virtual, mejorando la experiencia de usuario. La integración de los elementos interiores, como paredes divisorias, puertas y el piso texturizado, permite que el usuario recorra y visualice el espacio de manera intuitiva y realista. La disposición de estos detalles se realizó de manera que el modelo resultante sea compatible con las necesidades de renderizado en un sistema de realidad virtual, manteniendo un balance entre calidad visual y eficiencia computacional.

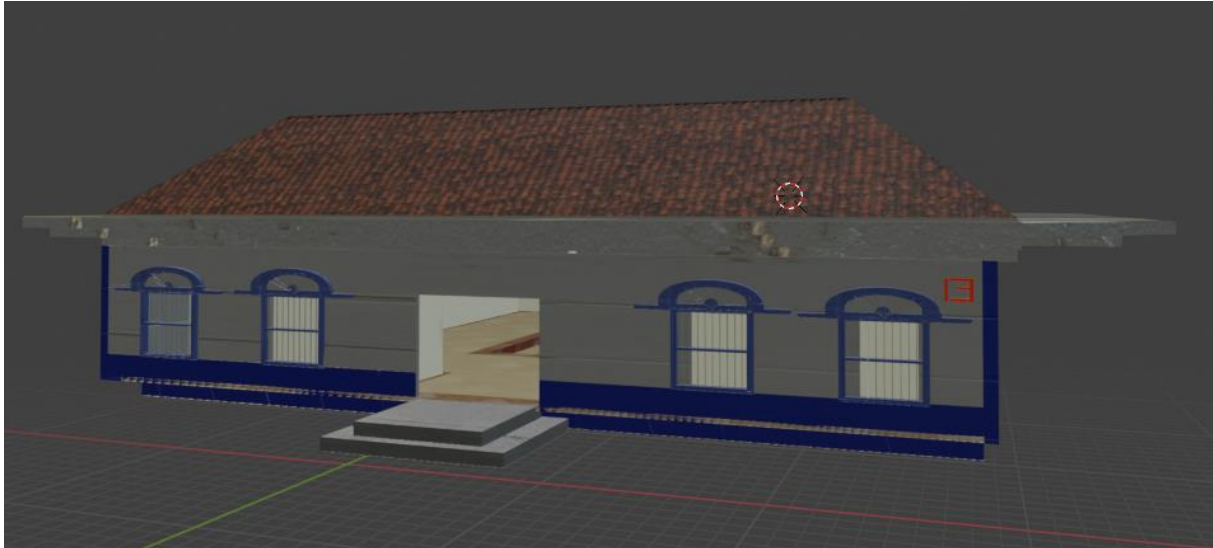


Figura 16.
Modelado final del edificio de secretaría en Blender

6.1.3.1. Desarrollo del entorno inmersivo con Unity.

- **Configuración del proyecto**

Una vez realizada la configuración de Unity descrita en la sección 5.3.1.4, permitió la creación de un entorno inmersivo compatible con Meta Quest 2 en Unity, esto integro de forma correcta la funcionalidad de los controles de realidad virtual y los lentes.

- **Integración de movimiento con la técnica Arm Swing**

Otro resultado de esta fase comprende integrar el sistema de movimiento del entorno virtual inmersivo, que permite a los usuarios navegar de forma cómoda dentro del mismo. Para ello, se evaluaron e integraron tres técnicas distintas de movimiento, la **Tabla 12** describe cada técnica, así como su explicación de funcionamiento y razón de descarte o aceptación.

Tabla 12.
Descripción de técnicas de movimiento

Técnica	Explicación	Uso
Teletransportación	Esta técnica permite a los usuarios moverse de forma instantánea de un punto a otro dentro del entorno, con el uso de los controles, se apunta y "dispara" en una superficie para moverse a ella.	Aunque fue efectiva y no presentaba mareos o malestar al ser probada, se descartó debido a que limitaba la inmersión, al no ser una experiencia de movimiento real o continua.
JoySticks	Mediante el uso de JoyStick de los controles de los Meta Quest 2, el usuario podía moverse en cualquier dirección.	La técnica resulto ser la más propensa a causar malestar o mareos, por lo cual fue descartada.
Arm Swing	Esta técnica simula el movimiento de balancear los brazos para generar	Esta técnica fue la seleccionada, debido a que

movimiento, lo que acerca más a la experiencia de caminar en el mundo real.

equilibra la comodidad de movimiento y la inmersión, además, reduce la posibilidad de causar malestar o mareos al sincronizar el movimiento físico con el movimiento virtual.

Entonces, la integración de la técnica de movimiento "Arm Swing" en entornos de realidad virtual es una solución efectiva para mejorar la inmersión y reducir el mareo o la incomodidad asociados con los desplazamientos virtuales.

Para integrarla en Unity, se utilizó un sistema basado en scripts personalizados junto con el XR Interaction Toolkit. Este enfoque incluyó la captura de los movimientos de los controladores de realidad virtual que emulan los movimientos de los brazos del usuario. En la **Figura 17** se muestra el script desarrollado, que detecta la amplitud y dirección de los movimientos de los controladores, permitiendo que el avatar del usuario se desplazará hacia adelante cuando el movimiento sea lo suficientemente significativo. Este método proporcionó un control de velocidad adaptable, lo que significa que un movimiento más rápido de los brazos resultó en un desplazamiento más acelerado dentro del entorno.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SwingingArmMotion : MonoBehaviour
{
    // Game Objects
    [SerializeField] private GameObject LeftHand;
    [SerializeField] private GameObject RightHand;
    [SerializeField] private GameObject MainCamera;
    [SerializeField] private GameObject ForwardDirection;

    //Vector3 Positions
    [SerializeField] private Vector3 PositionPreviousFrameLeftHand;
    [SerializeField] private Vector3 PositionPreviousFrameRightHand;
    [SerializeField] private Vector3 PlayerPositionPreviousFrame;
    [SerializeField] private Vector3 PlayerPositionCurrentFrame;
    [SerializeField] private Vector3 PositionCurrentFrameLeftHand;
    [SerializeField] private Vector3 PositionCurrentFrameRightHand;
```

```

//Speed
[SerializeField] private float Speed = 70;
[SerializeField] private float HandSpeed;

void Start()
{
    PlayerPositionPreviousFrame = transform.position; //Posicion actual
    PositionPreviousFrameLeftHand = LeftHand.transform.position; //Posicion anterior
    PositionPreviousFrameRightHand = RightHand.transform.position;
}

// Actualizacion en cada frame
void Update()
{
    // Obtener la dirección hacia adelante desde la cámara central y configurarla en el objeto
de dirección hacia adelante
    float yRotation = MainCamera.transform.eulerAngles.y;
    ForwardDirection.transform.eulerAngles = new Vector3(0, yRotation, 0);

    // Posicion de las manos
    PositionCurrentFrameLeftHand = LeftHand.transform.position;
    PositionCurrentFrameRightHand = RightHand.transform.position;

    // Posicion del jugador
    PlayerPositionCurrentFrame = transform.position;

    // Obtener la distancia recorrida por las manos
    var playerDistanceMoved = Vector3.Distance(PlayerPositionCurrentFrame,
PlayerPositionPreviousFrame);
    var leftHandDistanceMoved = Vector3.Distance(PositionPreviousFrameLeftHand,
PositionCurrentFrameLeftHand);
    var rightHandDistanceMoved = Vector3.Distance(PositionPreviousFrameRightHand,
PositionCurrentFrameRightHand);

    // Uso de velocidad de movimiento de las manos
    HandSpeed = ((leftHandDistanceMoved - playerDistanceMoved) +

```

```

(rightHandDistanceMoved - playerDistanceMoved));

    if(Time.timeSinceLevelLoad > 1f)
    {
        transform.position += ForwardDirection.transform.forward * HandSpeed * Speed *
Time.deltaTime;
    }

    // Establecer la posicion de las manos para el siguiente frame
    PositionPreviousFrameLeftHand = PositionCurrentFrameLeftHand;
    PositionPreviousFrameRightHand = PositionCurrentFrameRightHand;
    // Nueva posicion
    PlayerPositionPreviousFrame = PlayerPositionCurrentFrame;
}
}

```

Figura 17.

Script para la integración de Arm Swing

El script sincroniza los movimientos de los brazos con la cámara principal del usuario, garantizando que el desplazamiento ocurriera en la dirección en la que el usuario miraba. Se configuraron validaciones para asegurar que el movimiento se activará solo en momentos deseados, evitando interacciones involuntarias y garantizando una navegación fluida y controlada.

- **Implementación y Optimización del Skybox**

El uso de un SkyBox fue fundamental para simular un entorno envolvente, proporcionando un horizonte realista y mejorando la percepción de profundidad. Se seleccionó un SkyBox con texturas de alta calidad que imitan el cielo natural, añadiendo autenticidad al entorno. Además, se integraron Assets decorativos como sillas, césped e iluminación, tanto descargados desde la Asset Store de Unity como diseñados de forma personalizada en Blender. Estos elementos decorativos se distribuyeron estratégicamente para enriquecer el realismo del entorno y proporcionar detalles visuales que contribuyen a una experiencia inmersiva más completa.

- **Integración de modelos de Blender y Assets de Unity**

Para la migración de los modelos de Blender a Unity, los modelos se exportaron en formato FBX y se importaron con todas sus propiedades de textura y mapeo UV intactas. La **Figura**

18 evidencia el importe del modelo final del edificio dentro de Unity, además, se evidencian los detalles extra como la iluminación y los efectos visuales. Se empleó el sistema de materiales de Unity para afinar detalles y mejorar la estética de los objetos, aplicando sombreadores (shaders) compatibles con el renderizado para VR.



Figura 18.
Importe del modelo de edificio junto al SkyBox dentro de Unity

El terreno se creó en Unity usando la herramienta de Terrain, donde se aplicaron texturas de césped de alta calidad para dar una apariencia natural. Se añadieron detalles como variaciones de color y normal maps para realzar la profundidad y el realismo del césped. Las palmeras y otros elementos decorativos se incorporaron utilizando assets de la tienda de Unity, seleccionando aquellos que ofrecían modelos optimizados y texturas realistas.

La **Figura 19** se puede observar la distribución de las palmeras y otros detalles de vegetación, que fueron integrados de manera estratégica para no solo mejorar el aspecto visual sino también guiar intuitivamente al usuario durante la navegación. La iluminación ambiental y el uso de un Skybox tropical ayudaron a complementar el entorno, creando un espacio virtual acogedor y convincente. Estos elementos combinados proporcionaron una experiencia de recorrido inmersiva y más cercana a la realidad.



Figura 19.
Integración de edificio y elementos exteriores del edificio de secretaría dentro de Unity

La iluminación fue clave para destacar tanto la ambientación como la jugabilidad. Se implementaron luces direccionales para simular la luz solar, y luces de tipo spot y point para destacar objetos y zonas de interés. La combinación de luces dinámicas y horneadas permitió un balance entre calidad visual y rendimiento óptimo, evitando problemas de latencia y manteniendo los FPS (frames per second) estables en Meta Quest 2. Para acentuar los detalles y texturas, se emplearon sombras suaves y efectos de luz indirecta, elevando la calidad gráfica del entorno inmersivo y mejorando la interacción del usuario con el entorno.

Como resultado final, se anexa el código fuente generado en Unity para la creación del tour virtual inmersivo (Ver Sección **Anexo 1**).

6.1.4. Fase de implementación

El resultado de esta fase fue la integración y despliegue exitoso de la aplicación en los lentes de realidad virtual Meta Quest 2, logrando así poder ejecutar el proyecto nativamente en este dispositivo y no tener que asistirse de un computador o hardware externo, lo que además de facilitar el proceso de pruebas para los usuarios, también mejora la accesibilidad del programa, ya que el jugador solo debe usar los lentes de realidad virtual y podrá desplazarse por un espacio mucho más amplio ya que no estará conectado a ningún cable.

En la **Figura 20** se muestra la configuración para exportar el proyecto desde Unity hacia los Meta Quest 2.

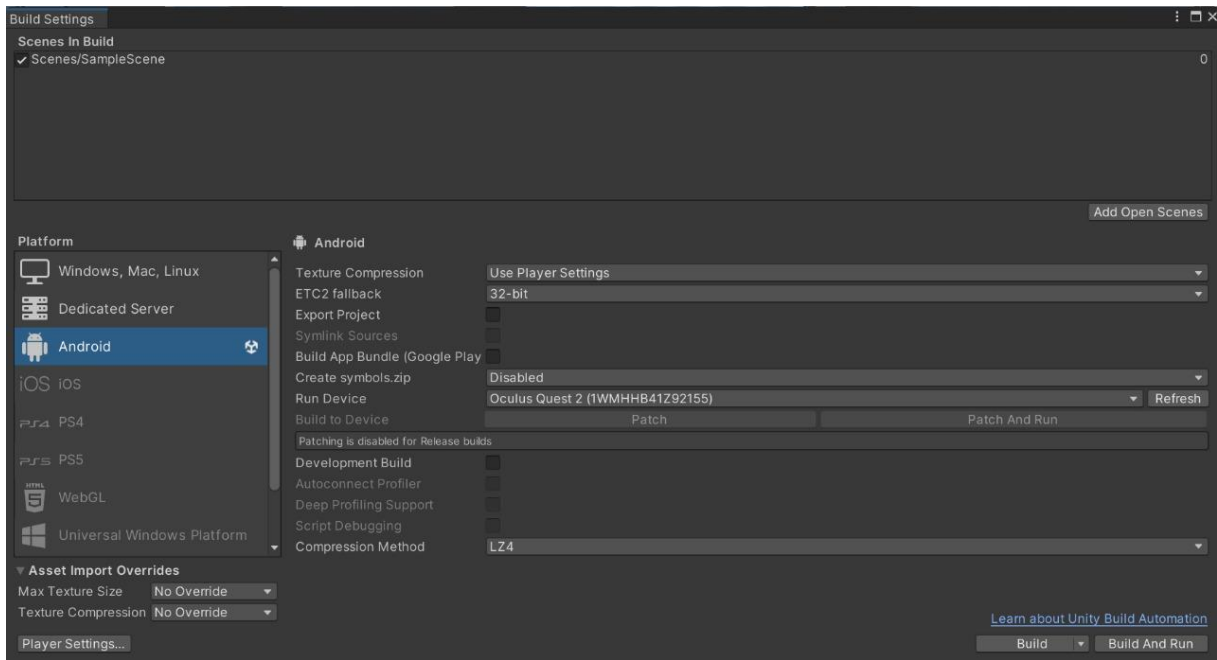


Figura 20.
Configuración para exportar entorno virtual a Meta Quest 2

En la **Figura 21** se puede apreciar la vista del menú de aplicativos de los Meta Quest 2, donde en primer lugar aparece el archivo APK de la aplicación con el nombre de “ENMACIS - Computación”, el cual fue otro de los resultados de esta fase, debido a que este modelo de lentes de realidad virtual está desarrollado con el sistema operativo de Android y por lo tanto para poder ejecutarlo nativamente en el hardware fue necesario generar un aplicativo compatible con dicho sistema, en este caso, se procedió a exportar la aplicación desde la plataforma de desarrollo utilizada, asegurando que el formato APK fuera adecuado y optimizado para el rendimiento de los lentes de realidad virtual.

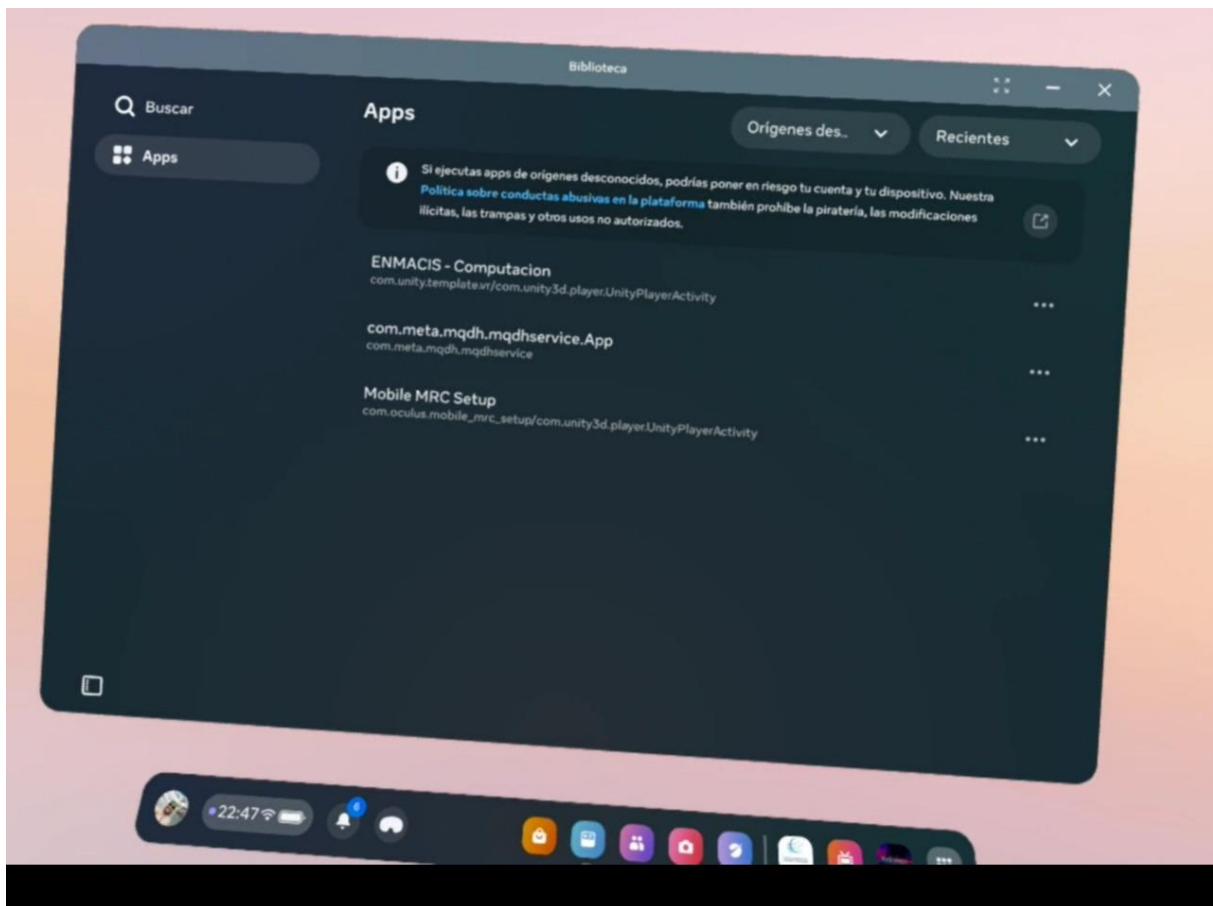


Figura 21.
Testeo de APK exportado e importado a los Meta Quest 2

Una vez generado el archivo APK, se realizó el proceso de instalación directa en los Meta Quest 2, lo que permitió verificar su funcionamiento en el dispositivo real. Esta integración no solo validó la ejecución del proyecto de manera autónoma, sino que también facilitó ajustes finales y pruebas del comportamiento de la aplicación en un entorno más realista, sin la dependencia de un equipo adicional.

6.1.5. Fase de pruebas

La **Tabla 13** recopila los comentarios y sugerencias brindadas por los usuarios que más retroalimentaron y probaron el tour. Además, dentro de estas sugerencias se han agrupado aquellas iguales o similares dadas por el resto de usuarios. También se incluye una “calificación” preliminar sobre la inmersión lograda por el entorno, que fue establecida en la escala de poco realista, realista y muy realista.

Tabla 13.*Resultado de las pruebas de usuario*

Usuario	Comentarios de retroalimentación	Calidad de inmersión
1	<ul style="list-style-type: none">- Mejora en el movimiento de "Arm Swing" modificando la sensibilidad con la que se detecta el movimiento de las manos.- Corrección en altura del personaje principal.- Cambio de texturas por unas de mayor calidad.	Realista
2	<ul style="list-style-type: none">- Mejora en las texturas generales del entorno.- Mejora en botones e indicaciones para realizar acciones (Por ejemplo, integrar un indicador cuando el usuario pueda hablar con el NPC).- Redimensión de elementos externos del edificio, como palmeras, arbustos, etc.	Muy realista
3	<ul style="list-style-type: none">- Implementar un nuevo SkyBox para representar el edificio administrativo de la facultad con mayor fidelidad.	Realista

6.2. Objetivo 2: Implementar un sistema de respuestas inteligentes y proactivas, con el modelo generativo "Gemini 1.5 Pro", para los avatares informativos dentro del sistema virtual inmersiva.

6.2.1. Fase de asignación

En esta fase se logró definir con claridad las herramientas tecnológicas y los componentes clave para la implementación del sistema de respuestas inteligentes. Para el lado del backend se optó por desarrollarlo con el entorno de ejecución de Javascript Node.js, junto con el framework de Express, debido a que estas opciones demostraron ser una elección adecuada por su capacidad para manejar solicitudes concurrentes de manera eficiente. Adicionalmente, se identificó que las APIs seleccionadas, las cuales fueron HuggingFace para la conversión de audio a texto y Azure Speech para la generación de audio, ya que estas dos herramientas eran compatibles con el acceso a ellas a través de solicitudes HTTP y ofrecían un rendimiento óptimo dentro del sistema diseñado.

6.2.2. Fase de análisis

En la **Figura 22** se muestra el diagrama de procesos obtenido como resultado de la fase de análisis, en el cual se detalla que representaron el proceso completo desde la detección del audio del usuario hasta la reproducción de la respuesta generada por los avatares. Estos diagramas permitieron identificar y mitigar posibles cuellos de botella en el sistema, optimizando el flujo de trabajo.

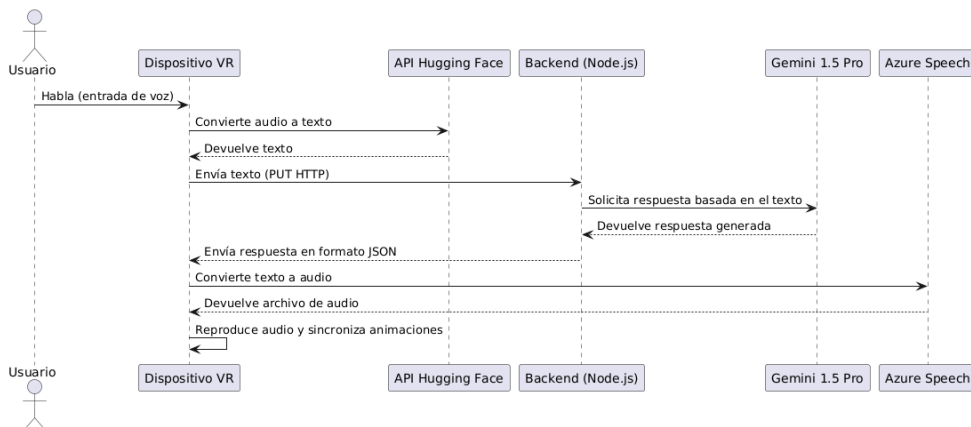


Figura 22.
Diagrama de procesos

Una vez con el diagrama de procesos definido, se tuvo una idea concisa sobre los límites del programa y a su vez el alcance de este, por lo cual se generó una base más estable sobre la cual trabajar al momento de proceder con la fase de construcción, ya que se tenía un conocimiento previamente delimitado sobre que funcionalidades se necesitarían y sobre todo

como se comunicarían e interactuarían cada uno de los sistemas o servicios involucrados en el desarrollo del entorno virtual

6.2.3. Fase de construcción

- **Implementación del BackEnd**

Para iniciar, se configuro el entorno para el desarrollo y posterior implementación del BackEnd. La **Figura 23** muestra la estructura de directorios utilizados.

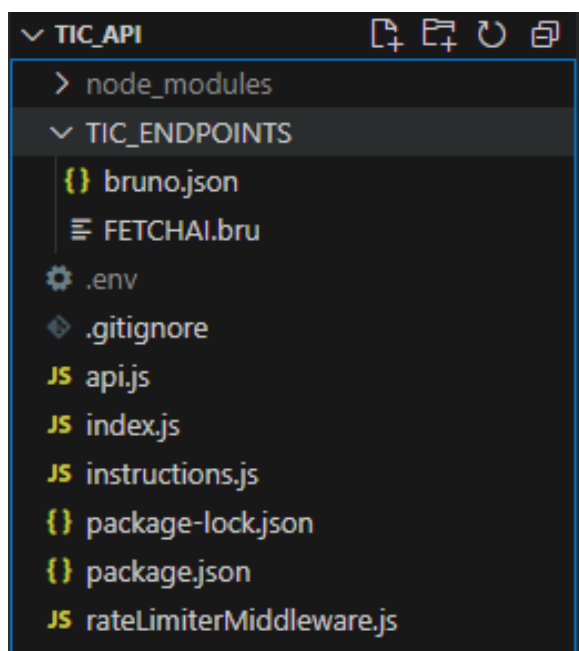


Figura 23.
Estructura de la ruta de archivos para el BackEnd

La **Tabla 14** muestra y describe el rol clave de cada archivo dentro del sistema BackEnd.

Tabla 14.
Descripción y función de cada archivo del BackEnd

Archivo	Funcionalidad
api.js	Define las configuraciones de red, incluyendo los encabezados CORS y los métodos de solicitud permitidos (GET y POST). Aunque inicialmente se consideraron solicitudes POST, se optó por el método GET para enviar preguntas como parámetros de la URL, ya que el procesamiento del cuerpo (body) en Unity al usar POST generaba retrasos significativos, afectando el rendimiento general.
index.js	Configura el servidor API y realiza la integración con el modelo Gemini 1.5 Flash , enviando las preguntas directamente a través de solicitudes GET. Este archivo también gestiona las respuestas retornadas por el modelo.
instructions.js	Contiene el prompt configurado para la IA, utilizando la documentación oficial de la Universidad Nacional de Loja (UNL) como base. Esto aseguró que las preguntas formuladas a la IA fueran procesadas con precisión en el contexto académico.
rateLimiterMiddleware.js	Implementa un limitador de tiempo para las solicitudes, previniendo sobrecargas en el servidor al restringir la frecuencia de acceso.

.env Contiene las variables de entorno necesarias para la configuración del sistema, incluyendo las credenciales de acceso a la API de Google Gemini.

Como resultado, la ejecución y construcción del Backend permitió responder correctamente a las solicitudes enviadas al endpoint principal, además, la configuración del `rateLimiterMiddleware` limitó eficazmente el número de solicitudes simultáneas, asegurando la estabilidad del sistema.

Además, las preguntas enviadas en formato GET fueron procesadas de manera eficiente por el modelo, devolviendo respuestas relevantes en formato JSON. La **Figura 24** muestra un ejemplo típico de respuesta a las solicitudes comunes.

```
{
  "status": "success",
  "data": {
    "responseText": "La respuesta generada basada en la documentación de la UNL."
  }
}
```

Figura 24.
Ejemplo de formato de respuesta a las solicitudes en JSON.

El modelo fue capaz de manejar preguntas complejas, gracias a la configuración del prompt en `instructions.js`. Se llevaron a cabo pruebas unitarias y de integración que validaron la robustez del sistema. Las respuestas se integraron correctamente al entorno virtual, permitiendo una experiencia interactiva fluida, además, de generar mayor inmersión debido al procesamiento de la respuesta para ser integrada mediante una voz.

Estos scripts permitieron a los avatares detectar la presencia del usuario, activar el micrófono, enviar el audio capturado al backend y reproducir el audio de respuesta sincronizando las animaciones labiales. Además, se integraron sistemas de validación para garantizar que las interacciones fueran activadas únicamente cuando el usuario estuviera dentro del rango de interacción del avatar.

- **Uso del API de HuggingFace**

Para poder tener acceso al API de HuggingFace y por lo tanto a sus diferentes modelos o funcionalidades, fue necesario crear una cuenta y de igual manera un `API_KEY` para posteriormente conectarla con Unity. En la **Figura 25** se puede apreciar una cuenta creada dentro de la plataforma web de HuggingFace, junto con una key solicitada específicamente para el desarrollo de este trabajo de integración curricular.

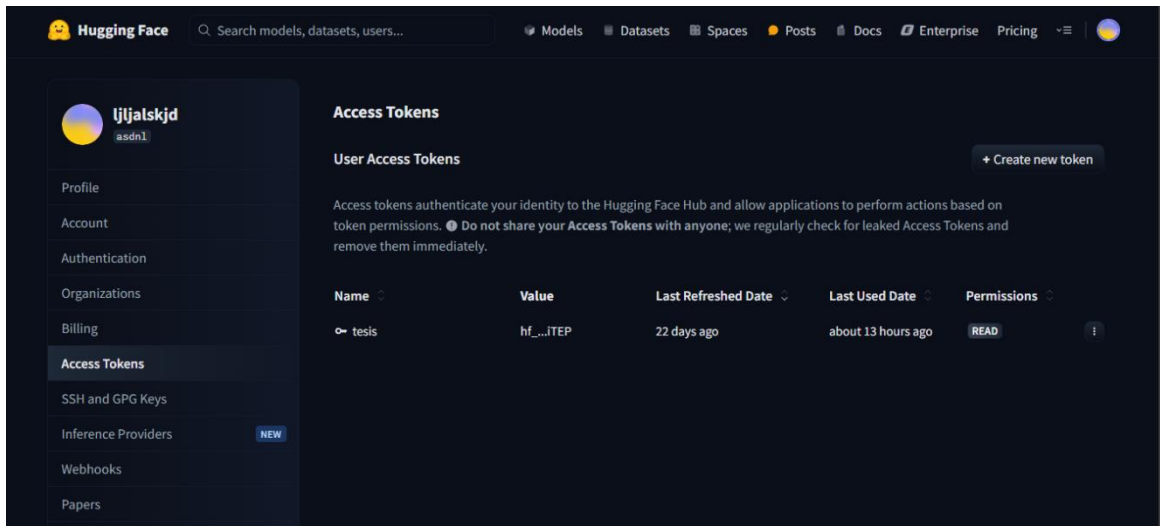


Figura 25.
Cuenta creada y validada de Hugging Face

Para poder conectar este API a Unity y poder interactuar con sus endpoints, se hizo uso de la librería de HuggingFace que proporciona Unity a través de su AssetStore, una vez descargada se accede a la ventana de HuggingFace y se coloca el ApiKey que se obtuvo previamente. En la **Figura 26** se muestra cómo se encuentra configurado la biblioteca de este servicio dentro de Unity

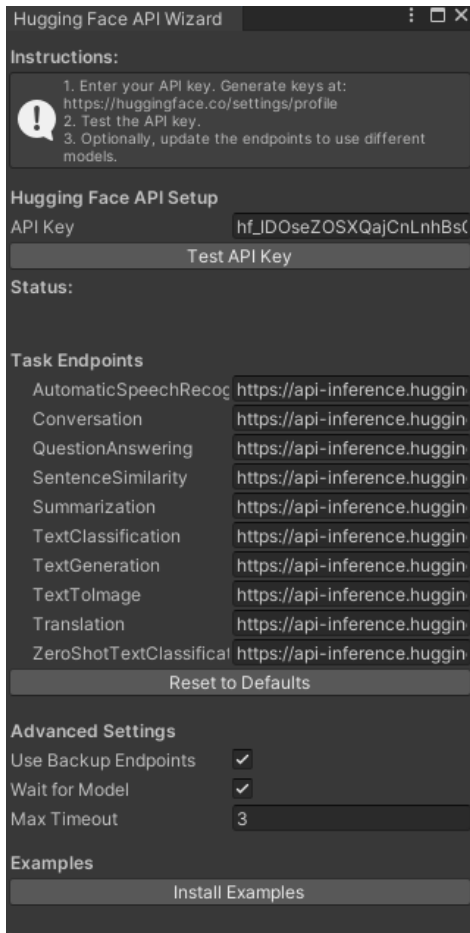


Figura 26.
Configuración de la biblioteca de HuggingFace dentro de Unity

Por último, se definió el script el cual se encargaría de reconocer cuando el usuario este hablando y esperar a que termine para procesarlo y convertirlo en texto mediante el api. En la **Figura 27** se muestra el método `IniciarReconocimientoVoz`, el cual es el encargado de llamar al script que va a reconocer la voz del usuario, para esto primero limpia el texto que el usuario haya generado previamente, reinicia la bandera de que el usuario ha terminado de hablar y comprueba que el botón para poder preguntar este presionado y a su vez que no se esté realizando ningún tipo de solicitud, finalmente llama al método de `StartRecording` y espera a que el jugador termine de hablar para enviar el texto generado. Además, también se encuentra el método `StartRecording`, en el cual se comprueba que el jugador este dentro del trigger y en caso de ser así, se activa el micrófono para captar el audio y se activa la bandera de recording.

```
public IEnumerator IniciarReconocimientoDeVoz()
{
    playerText = string.Empty;
```

```

playerFinishedTalking = false;
Debug.Log("Se va a comprobar si el boton esta presionado");
if (isPressedButton) {
    if (isFetching) {
        ChangeText("Espera tu respuesta para poder volver a preguntar");
        yield return null;
    }
    dictationScript.StartRecording();

    yield return new WaitUntil(() => playerFinishedTalking);

    StartCoroutine(HacerSolicitud());
}
}
public void StartRecording()
{
    if (triggerActive.triggerUsed) {
        triggerActive.ChangeText("Hablando...., suelta el boton para dejar de hablar");
        clip = Microphone.Start(null, false, 30, 44100);
        recording = true;
    }
}
}

```

Figura 27.

Script de Unity en C# para los métodos IniciarReconocimientoVoz y StartRecording

Por ultimo al momento de que el usuario termine de hablar, el audio obtenido se convierte en arreglo de bytes mediante el cual el api de HuggingFace se encarga de procesar y convertir en texto, En la **Figura 28** se puede ver la lógica del método StopRecording, en donde se comprueba si se está grabando el audio y si el micrófono funciona correctamente, en caso de ser así se llama al método sendRecording, y es aquí donde se llama al api de conversión de audio a texto, se indica que el jugador termino de hablar y se devuelve el texto generado.

```

public void StopRecording()
{
    if (recording)
    {
        triggerActive.ChangeText("Terminaste de hablar");
    }
}

```

```

if (Microphone.IsRecording(null))
{
    var position = Microphone.GetPosition(null);
    Microphone.End(null);

    if (clip != null && position > 0 && position <= clip.samples)
    {
        var samples = new float[position * clip.channels];
        clip.GetData(samples, 0);
        bytes = EncodeAsWAV(samples, clip.frequency, clip.channels);
        SendRecording();
    }
    else
    {
        Debug.Log("Clip no válido o posición fuera de rango.");
    }
}
else
{
    Debug.Log("");
}

recording = false;
}
}
private void SendRecording()
{
    HuggingFaceAPIAutomaticSpeechRecognition(bytes, response =>
    {
        triggerActive.ChangeText("Pensando la respuesta");
        Debug.Log("El jugador ha dejado de hablar: " + response);
        triggerActive.SetPlayerFinishedTalking(response);
    }, error =>
    {
        triggerActive.ChangeText("Error");
    });
}
}

```


Figura 28.

Script de Unity en C# para los métodos StopRecording y SendRecording

- **Configuración de Azure Speech**

Para configurar Azure se accedió con una cuenta que posee un correo universitario, para de esta forma aprovechar el convenio de la universidad con la plataforma. En la **Figura 29** se muestra cómo se configuro el servicio de Speech para que funcionara de acuerdo con las medidas y necesidades correctas de este proyecto.

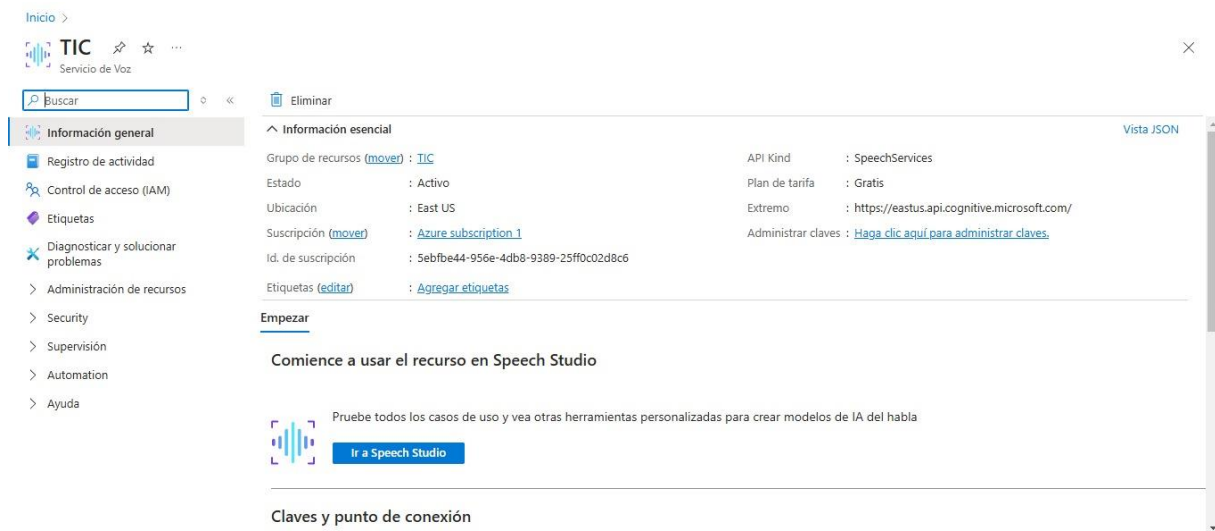


Figura 29.

Configuración del servicio Azure Speech

Para poder conectar este servicio a Unity fue necesario implementar un método que se conecte de forma correcta con el api de Azure y que a su vez permita determinar con que voz se desea que se genere el audio y que texto es el que se procederá a convertir. En la **Figura 30** se puede ver la lógica del método SendTextToSpeechRequest, donde se envía una solicitud a la API de Azure para convertir texto en audio. Primero, se configura la solicitud HTTP, estableciendo los encabezados necesarios, como la clave de suscripción y el formato de salida en riff-16khz-16bit-mono-pcm (WAV).

Luego, se construye el contenido en formato SSML (Speech Synthesis Markup Language), que define el texto a convertir junto con la voz seleccionada. El cuerpo de la solicitud es codificado en UTF-8 y asignado a un UploadHandlerRaw, mientras que un DownloadHandlerBuffer se encarga de recibir la respuesta. La petición se envía y, en caso de éxito, el audio generado se almacena en un archivo WAV dentro del directorio persistente de la aplicación. Posteriormente, el método LoadAudioFromFile carga el archivo de audio utilizando UnityWebRequestMultimedia.GetAudioClip, lo convierte en un AudioClip, lo asigna

al audioSource y lo reproduce. Una vez que la reproducción termina, el archivo es eliminado del sistema para liberar espacio. Si la carga del audio falla, se muestra un mensaje de error en la consola.

```
private IEnumerator SendTextToSpeechRequest(string text, string voiceName)
{
    // Configuración de la solicitud TTS
    Debug.Log("Texto que se va a enviar a Azure: " + text);
    var request = new UnityWebRequest(ttsEndpoint, "POST");
    request.SetRequestHeader("Ocp-Apim-Subscription-Key", subscriptionKey);
    request.SetRequestHeader("Content-Type", "application/ssml+xml");
    request.SetRequestHeader("X-Microsoft-OutputFormat", "riff-16khz-16bit-mono-pcm"); //
Usar WAV

    // Generar el contenido SSML para el TTS de Azure
    string ssml = $"@
<speak version='1.0' xmlns='http://www.w3.org/2001/10/synthesis' xml:lang='en-US'>
    <voice name='{voiceName}'>{text}</voice>
</speak>";
    Debug.Log(ssml);
    byte[] bodyRaw = Encoding.UTF8.GetBytes(ssml);
    request.uploadHandler = new UploadHandlerRaw(bodyRaw);
    request.downloadHandler = new DownloadHandlerBuffer();

    // Enviar la solicitud y esperar respuesta
    yield return request.SendWebRequest();

    if (request.result != UnityWebRequest.Result.Success)
    {
        Debug.LogError("Error en la solicitud TTS: " + request.error);
    }
    else
    {
        Debug.Log("Respuesta de la API: " + request.downloadHandler.text);
        // Guardar el archivo WAV en disco
        byte[] audioData = request.downloadHandler.data;
        string filePath = Path.Combine(Application.persistentDataPath, "tts_audio.wav");
```

```

File.WriteAllBytes(filePath, audioData);

// Cargar el archivo WAV en Unity
yield return StartCoroutine(LoadAudioFromFile(filePath));
}
}
}

private IEnumerator LoadAudioFromFile(string filePath)
{
    using (UnityWebRequest audioLoader = UnityWebRequestMultimedia.GetAudioClip("file://"
+ filePath, AudioType.WAV))
    {
        yield return audioLoader.SendWebRequest();

        if (audioLoader.result == UnityWebRequest.Result.Success)
        {
            AudioClip audioClip = DownloadHandlerAudioClip.GetContent(audioLoader);
            audioSource.clip = audioClip;
            audioSource.Play();

            // Esperar hasta que el audio termine de reproducirse
            yield return new WaitUntil(() => !audioSource.isPlaying);

            // Eliminar el archivo después de la reproducción
            if (File.Exists(filePath))
            {
                File.Delete(filePath);
                Debug.Log("Archivo de audio eliminado: " + filePath);
            }
        }
        else
        {
            Debug.LogError("Error cargando el audio: " + audioLoader.error);
        }
    }
}

```

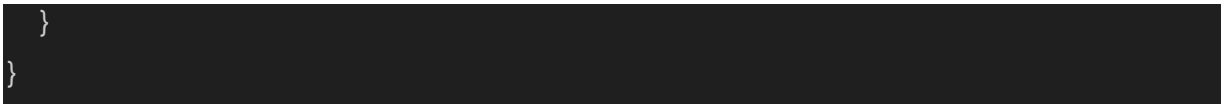


Figura 30.

Métodos LoadAudioFromFile y SendTextToSpeechRequest

- **Integración de Funcionalidades en Unity con N.P.C.s**

Una vez con el backend configurado y preparado para recibir preguntas por parte del usuario, se procedió a implementar la lógica para que los npcs reconozcan cuando el jugador se encuentra cerca de ellos y cuando presionen el botón para hablar.

En primer lugar, era necesario tener un modelo visual el cual los jugadores pudieran ver y supiesen que son estos los que solventarían sus dudas, por lo mismo para obtener un modelo que represente a un encargado u encargada de una oficina de secretaria, se usó el servicio de Mixamo, el cual ofrece modelos 3D de manera gratuita, y a su vez permite animar estos de forma sencilla. En la **Figura 31** se encuentra el modelo seleccionado en la plataforma web y a su vez los que serán utilizados para representar a los NPC'S dentro del entorno virtual.

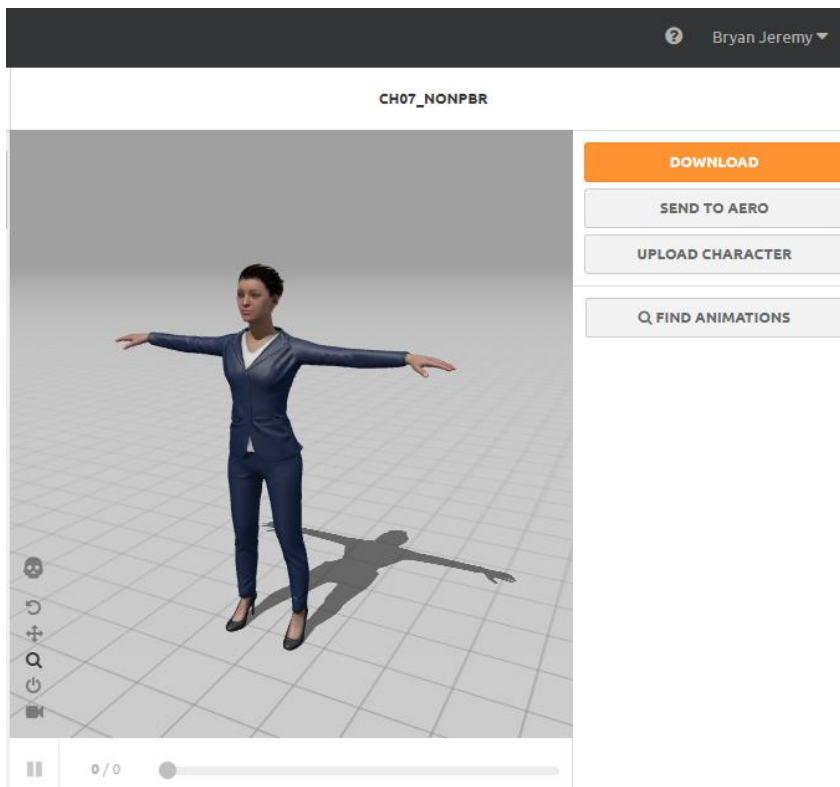


Figura 31.

Modelo 3D seleccionado en Mixamo

Posteriormente a cada uno de los modelos, los cuales fueron asignados en sus correspondientes oficinas, se le colocó un collider de tipo box o caja, que permite detectar a un jugador cuando este está cerca de la secretaria y además si puede hablar o no con ella. En la **Figura 32** se muestra cómo estaba configurado el tamaño de los colliders, donde estos tienen

un alto y ancho parecido al de los modelos, pero con un poco más de longitud para que el jugador no tenga que estar demasiado aproximado para poder interactuar

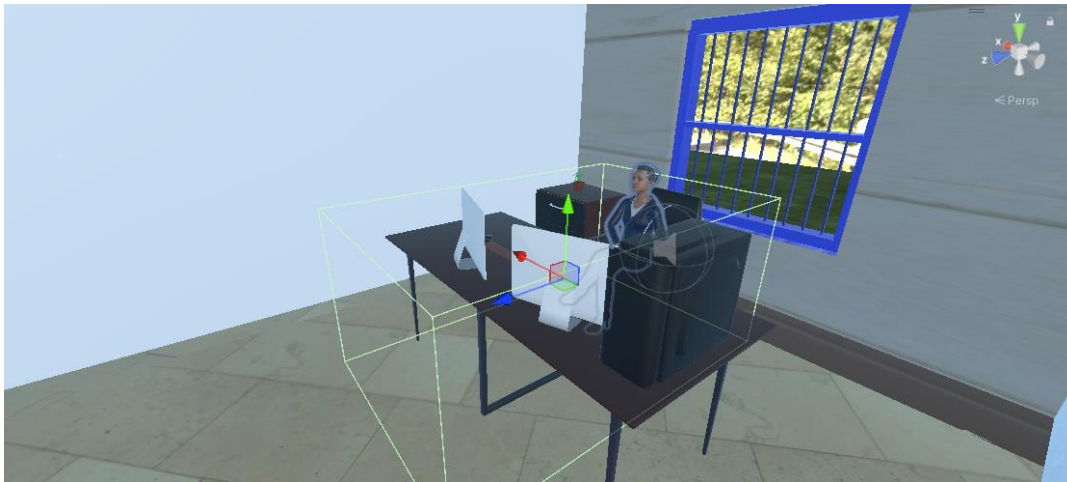


Figura 32.
Collider de Unity para la detección del jugador

Para detectar cuando un jugador está cerca de una secretaria, se configuro mediante las propiedades de cada uno de estos Colliders, que sean de tipo “Trigger”, es decir que cuando un jugador se encuentre dentro de ellos, procedan a disparar un evento que posteriormente se recuperara mediante código. En la **Figura 33** se muestra cómo se encuentran asignadas las propiedades de dichos Colliders dentro de las configuraciones del motor grafico de Unity.

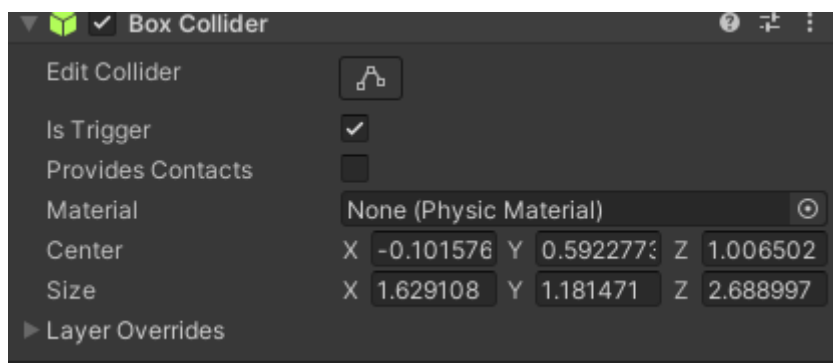


Figura 33.
Configuración de Collider dentro de Unity

A través del script programado, se recupera la referencia a cada uno de estos Colliders y se llama al evento en caso de que algo entre dentro de los límites de este, en este caso al solo tener como interés que se ejecute un evento únicamente cuando lo que entra en los límites de este objeto es un jugador, se agregó el condicional que indique esta especificación.

En la **Figura 34** se muestra como el método de `OnTriggerEnter`, el cual es un método por defecto de Unity, tiene la limitante de que solo va a ejecutar sus líneas lógicas siempre y cuando el Tag del objeto que entre al mismo sea de tipo `Player`, o por su traducción al español jugador. En el caso de que sea correcta esta condicional, se procede a mostrar un texto para el usuario indicando que debe presionar el botón de B para poder hablar, a su vez, la bandera de `triggerUsed` cambia su estado a `True` para indicar al resto del sistema que el jugador está dentro del `Collider` y que por lo tanto va a poder hablar, finalmente se activa el método de `StartTalking` que inicia la conversación con el NPC.

```
private void OnTriggerEnter(Collider other)
{
    if (other.CompareTag("Player") && !triggerUsed) // Solo activa si es el jugador y no se ha
    usado el trigger
    {
        Debug.Log("¡El jugador ha activado el trigger!");
        ChangeText("Presiona B para hablar");
        triggerUsed = true; // Marca el trigger como usado
        StartCoroutine(ActivarTalking());
    }
}
```

Figura 34.
Método `OnTriggerEnter` para detectar la presencia del jugador

Posteriormente se asignó la lógica de la ejecución de un evento cuando el usuario presiona el botón correspondiente. En la **Figura 35** se muestra cómo se asigna la función de `StartRecording` mientras se mantenga presionado el botón secundario de Acción, que en este caso es el botón B del controlador derecho de los Meta Quest, de igual forma también se puede apreciar como la lógica del método mencionado comprueba que el jugador este dentro del `collider` a través de la bandera que se definió en el script anterior, en caso de ser así, se bloquea el movimiento de `ArmSwing`, se activa la bandera de que el botón está presionado y se ejecuta el evento para reconocer la voz a través del Api de `HuggingFace` que se detallara más adelante.

```
private void OnEnable() {
    secondaryButtonAction.action.started += StartRecording;
    secondaryButtonAction.action.canceled += StopRecording;
    secondaryButtonAction.action.Enable(); // Asegúrate de habilitar el Input Action
}
```

```

}

private void OnDisable() {
    secondaryButtonAction.action.started -= StartRecording;
    secondaryButtonAction.action.canceled -= StopRecording;
    secondaryButtonAction.action.Disable();
}

private void StartRecording(InputAction.CallbackContext context) {
    Debug.Log("Botón B presionado - Iniciando grabación:" + TriggerActive.triggerUsed);
    if (TriggerActive.triggerUsed) {
        swing.enabled = false;
        Debug.Log("Se esta grabando voz");
        TriggerActive.isPressedButton = true;
        StartCoroutine(TriggerActive.IniciarReconocimientoDeVoz());
    }
}
}

```

Figura 35.

Código de Método StartRecording y asignación a botón de controles físicos

Una vez que se reconoció lo que el usuario preguntó y se convirtió en texto, se procede a enviarlo al API conectada con el modelo de Gemini, para esto se hace la solicitud correspondiente, enviando a través del body de la consulta el texto o pregunta generada por el usuario. En la **Figura 36** se muestra la lógica del método encargado de comunicarse con el API desarrollada, para esto primero comprueba que no se esté haciendo una solicitud a través de la bandera isFetching, después utiliza los métodos proporcionados por Unity para enviar el método HTTP de tipo PUT y enviar el texto generado, una vez que la respuesta fue recibida, se procede a llamar al método de conversión de texto a audio mediante Azure e igualmente reproduciéndolo, por último se muestra en texto la respuesta que se generó y se resetean las banderas para poder volver a realizar otra pregunta.

```

private IEnumerator HacerSolicitud()
{
    if (isFetching) yield return null;
    isFetching = true;
    UnityWebRequest request = UnityWebRequest.Get(url + playerText);
    Debug.Log(request.url);
    yield return request.SendWebRequest();
}

```

```

if (request.result == UnityWebRequest.Result.ConnectionError || request.result ==
UnityWebRequest.Result.ProtocolError)
{
    Debug.LogError("Error: " + request.error);
    ChangeText("Error en la solicitud: " + request.error);
}
else
{
    string jsonResponse = request.downloadHandler.text;
    Debug.Log("Respuesta de la API: " + jsonResponse);
    ApiResponse response = JsonUtility.FromJson<ApiResponse>(jsonResponse);
    animator.SetBool("IsTalking", true);
    azureTTS.SynthesizeAndPlay(response.message, voice);
    StopCoroutine(HacerSolicitud());
    playerText = "";
    ChangeText(response.message);
    isFetching = false;
    yield return new WaitForSeconds(1.0f);
    animator.SetBool("IsTalking", false);
}
}

```

Figura 36.

Código para realizar la solicitud

6.2.4. Fase de implementación

El backend fue desplegado en un servidor en la nube, asegurando alta disponibilidad y escalabilidad para soportar múltiples usuarios simultáneamente. Los ajustes realizados en Unity permitieron sincronizar perfectamente las respuestas generadas por el backend con las animaciones de los avatares y la reproducción de voz.

Se calibraron los parámetros de Azure Speech para garantizar la calidad y naturalidad del audio generado.. Esto resultó en una percepción más inmersiva y realista del entorno virtual.

6.2.5. Fase de pruebas

- **Selección de usuarios**

Para las pruebas realizadas, se tomó estudiantes de primer y quinto ciclo de la Carrera de Computación, que interactuaron con el entorno virtual no inmersivo y con el entorno

inmersivo, con el fin de comparar de forma más eficiente el cambio en la SoP. La muestra obtenida fue de 39 estudiantes (Los resultados de los cuestionarios son accesibles en la sección **Anexo 6**).

- **Explicación y pruebas**

Una vez seleccionados los cursos, se realizó una breve explicación del proyecto, su propósito y las pruebas respectivas con los usuarios, además, se explicó la forma de interactuar con los NPC, el movimiento dentro del entorno y demás información relevante para poder interactuar de forma correcta con el (Ver Sección **Anexo 8**).

- **Aplicación del test SUS**

Una vez realizadas las pruebas, los usuarios respondieron al test de SoP, la **Tabla 15** resume los resultados obtenidos en cada ítem del test, así como la desviación típica de cada pregunta calculada como se observa en (2).

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}} \quad (2)$$

σ = Desviación típica

x_i = Respuesta a cada ítem

\bar{x} = Promedio de respuestas

N = Numero de respuestas

Tabla 15.

Resultado de aplicación de test SuS

Item	Promedio	Desviación típica
1	6.41	0.63
2	5.90	1.01
3	6.13	1.09
4	6.38	0.7
5	6.36	0.62
6	6.36	0.62

- **Alfa de Cronbach y promedio final**

Las pruebas realizadas con el cuestionario SUS (System Usability Scale) arrojaron datos cuantitativos valiosos sobre la experiencia de usuario en el sistema inmersivo. Para medir y corroborar la fiabilidad de los resultados obtenidos se realizó el cálculo del alfa de Cronbach con la fórmula mostrada en (3).

$$\alpha = \frac{K}{K-1} \left[1 - \frac{\sum S_i^2}{S_T^2} \right] \quad (3)$$

α = Alfa de Cronbach

K = Número de ítems

S_i = Varianza de cada ítem

S_T = Varianza del total

Sabiendo que la varianza de cada ítem se calcula como (4):

$$\sigma_i^2 = \frac{\sum(x-\bar{x})^2}{n} \quad (4)$$

Se puede realizar el cálculo del alfa de Cronbach

$$\alpha = \frac{6}{6-1} \left[1 - \frac{(4.17)^2}{(13.49)^2} \right]$$
$$\alpha = 0.83$$

Entonces, el nivel de coeficiente del Alfa de Cronbach es de 0.83, y sabiendo que mientras este valor se aproxima más a 1 es más fiable [14], se puede determinar el promedio final de la SoP percibida.

Conociendo los resultados de la **Tabla 15**, podemos calcular el promedio final mediante (5) como se muestra a continuación:

$$SoP = \frac{\sum Mi}{N} \quad (5)$$

SoP = Sensación de presencia percibida

Mi = Media de respuestas para cada ítem

N = Número de ítems

$$SoP = \frac{37.54}{6}$$

$$SoP = 6.26$$

Para facilitar la comparativa, el valor de SoP puede ser representado de forma porcentual mediante (6):

$$P = \frac{SoP}{E} * 100 \quad (6)$$

P = Porcentaje de SoP

E = Escala

Aplicando el máximo de la escala utilizada (de Likert), que es 7, obtenemos que:

$$P = \frac{6.26}{7} * 100$$

$$P = 89.43\%$$

Sabiendo esto, la **Tabla 16** resume el resultado comparativo de la SoP percibida antes y después de actualizar el entorno virtual no inmersivo de la carrera a un entorno inmersivo.

Tabla 16.
Resultado de SoP en entorno no inmersivo y entorno inmersivo

Entorno virtual	SoP	SoP (%)
No inmersivo [1]	6.01	86
Inmersivo	6.26	89.43
Diferencia	0.25	3.43

Nota: SoP (Sensación de Presencia)

Finalmente, según la información teórica [2], [47], [48], [49], establece que si se supera el punto medio de la escala (3.5 usando la escala de Likert), indica una notable SoP percibida por parte de los usuarios que interactuaron con el entorno virtual.

7. Discusión

7.1 Primer objetivo:

El primer objetivo de esta investigación consistió en actualizar el sistema de tour virtual de la carrera de Computación de la FEIRNNR a un entorno inmersivo mediante la metodología MSD-VR. Los resultados obtenidos demostraron la viabilidad y efectividad de esta actualización, logrando desplegar un entorno virtual en Unity que fue ejecutado exitosamente en los dispositivos Meta Quest 2.

La actualización de un entorno virtual no inmersivo a uno inmersivo ha sido estudiada en la literatura. Autores como Slater y Wilbur [16] destacan que la inmersión es un factor clave para aumentar la sensación de presencia en entornos virtuales, por otro lado, la inmersión aumenta mediante el uso de tecnologías como los visores de realidad virtual [2]. En este sentido, los resultados de este estudio son concordantes con la literatura previa, ya que la implementación del entorno inmersivo en Meta Quest 2 permitió una experiencia más envolvente y realista en comparación con el sistema no inmersivo anterior.

Sin embargo, es importante destacar que algunos estudios, como el de Cummings y Bailenson [2], señalan que la efectividad de la inmersión depende en gran medida de la calidad de los modelos 3D y la fluidez del sistema. En este proyecto, se utilizaron modelos 3D optimizados y técnicas de renderizado avanzadas en Unity, lo que contribuyó a un rendimiento fluido y una experiencia de usuario satisfactoria. Esto corrobora la importancia de una planificación técnica rigurosa en la actualización de entornos virtuales.

La metodología MSD-VR [9] demostró ser una herramienta efectiva para guiar el proceso de actualización del tour virtual. Esta metodología, se enfoca en el diseño sistemático de aplicaciones de realidad virtual, abarcando desde la asignación de recursos hasta la implementación y pruebas. En este estudio, la metodología permitió estructurar el desarrollo del entorno inmersivo en fases claras y bien definidas, lo que facilitó la identificación y corrección de errores en etapas tempranas.

En relación con la pregunta de investigación, los resultados obtenidos en este objetivo específico proporcionan evidencia preliminar de que la actualización a un entorno inmersivo contribuye positivamente a la sensación de presencia. Aunque la medición cuantitativa de la sensación de presencia se abordó en el segundo objetivo, la implementación exitosa del entorno inmersivo sentó las bases para una experiencia más envolvente, lo cual es un requisito previo para aumentar la sensación de presencia.

Una limitación de este objetivo fue la dependencia de los recursos técnicos disponibles, como los dispositivos Meta Quest 2 y las capacidades de renderizado de Unity. Aunque estos recursos fueron suficientes para lograr un entorno inmersivo funcional, futuras

investigaciones podrían explorar el uso de tecnologías más avanzadas, como realidad aumentada o sistemas de seguimiento ocular, para mejorar aún más la experiencia del usuario.

7.2 Segundo objetivo

En este objetivo, se implementó un sistema de respuestas inteligentes y proactivas utilizando la IA generativa "Gemini 1.5 Pro", destinado a mejorar la interacción de los usuarios con los avatares informativos dentro del entorno virtual inmersivo. A través de esta implementación, se buscaba generar una experiencia más fluida y natural, donde los usuarios pudieran obtener respuestas contextuales y pertinentes mientras exploraban el entorno virtual.

No obstante, una limitación observada fue en el uso de la metodología MSD-VR [9], debido a la necesidad de ajustar algunos aspectos para adaptarla a las particularidades del proyecto, como la integración de respuestas inteligentes y la optimización para dispositivos móviles como los Meta Quest 2. Esto sugiere que, aunque la MSD-VR es una metodología robusta, puede requerir adaptaciones dependiendo del contexto y los recursos disponibles.

Como se observa en la **Figura 37**, los resultados obtenidos en la **Tabla 16**, medidos mediante los test de sensación de presencia, permitieron contestar la pregunta de investigación plantada: **¿En qué medida la mejora del sistema de tour virtual no inmersivo, de la carrera de Computación de la Facultad de la Energía las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja, aumenta la sensación de presencia?**, cuyas respuestas mostraron una mejora del 3.43% en comparación con el sistema previo basado en tecnologías no inmersivas.

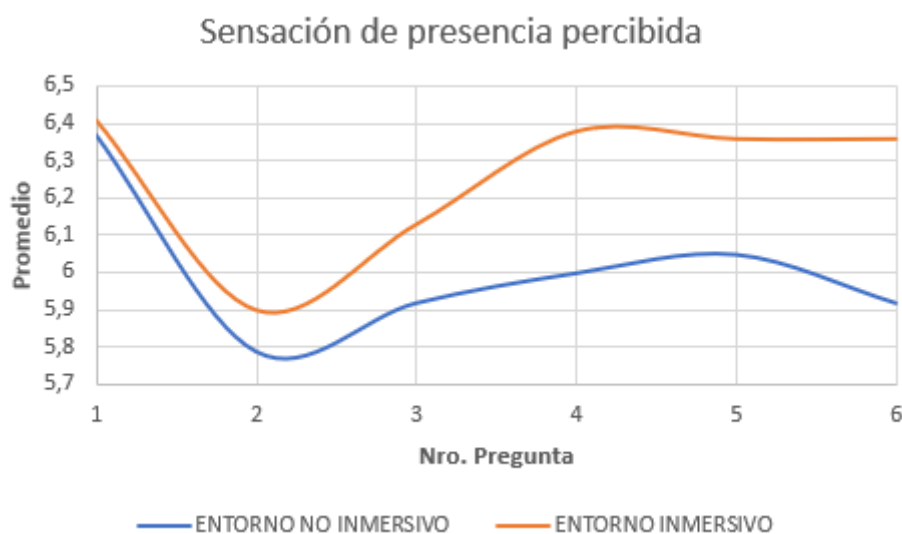


Figura 37.

Representación gráfica de los resultados comparativos en la SoP percibida en ambos entornos

Este incremento es relevante dentro del contexto de la investigación, pues indica que el uso de inteligencia artificial generativa y el uso de lentes de realidad virtual pueden efectivamente contribuir a mejorar la percepción de los usuarios respecto a la presencia en entornos virtuales.

A pesar de haber aumentado porcentualmente, los resultados obtenidos en este estudio son ligeramente menores al esperado, esto podría deberse a varias razones, como las características específicas de la muestra, que pueden haber influido en la percepción de los participantes.

La pregunta central sobre si la mejora del sistema aumentaría la SoP percibida en los usuarios se respondió afirmativamente. La implementación de respuestas inteligentes contribuyó positivamente, pero aún existen áreas de mejora, particularmente en la interacción fluida y la precisión de las respuestas generadas por la IA.

Una de las principales limitaciones de este estudio es el tamaño y las características de la muestra utilizada, que puede no ser completamente representativa de la población general. Además, el entorno virtual no completamente inmersivo pudo haber sido limitado debido a la falta de uso de IA, que puede afectar a la SoP. Futuros estudios podrían ampliar la muestra y explorar la implementación de tecnologías de realidad virtual completa para evaluar su impacto en la presencia.

8. Conclusiones

Este estudio ha permitido validar la efectividad de la mejora del sistema de tour virtual de la carrera de Computación de la Facultad de la Energía las Industrias y los Recursos Naturales no Renovables de la Universidad Nacional de Loja mediante la incorporación de tecnologías inmersivas y la implementación de un sistema de respuestas inteligentes con IA generativa.

En relación con el objetivo general alcanzado, la actualización del sistema a un entorno inmersivo utilizando la metodología MSD-VR permitió desarrollar un modelo funcional desarrollado en Blender, integrado en Unity y desplegado en los dispositivos Meta Quest 2. Este avance proporciona una plataforma sólida para futuras interacciones inmersivas, aunque se identificaron áreas de mejora, especialmente en la fluidez de las interacciones entre los usuarios y el entorno.

El aumento en la SoP fue de un 3.43% y, aunque esta mejora es significativa, se reconoce que los resultados podrían haberse visto potenciados con un entorno completamente inmersivo y un ajuste adicional en la precisión y contexto de las respuestas generadas por la IA.

Concluyendo, los hallazgos de este Trabajo de Integración Curricular demuestran que las tecnologías inmersivas y los sistemas inteligentes pueden mejorar la experiencia del usuario en entornos virtuales, aunque queda espacio para optimizar estos sistemas en el futuro. Los resultados obtenidos validan parcialmente las hipótesis planteadas, sugiriendo que la implementación de estas tecnologías puede tener un impacto positivo en la SoP de los usuarios, pero también revelan las limitaciones inherentes a la tecnología actual y las características del entorno utilizado.

El estudio abre la puerta para futuras investigaciones que puedan profundizar en la integración de realidad virtual completa, optimización de la IA generativa y ampliación de las pruebas en entornos educativos y profesionales.

9. Recomendaciones

Se recomienda continuar perfeccionando el modelo inmersivo utilizado en la actualización del sistema de tour virtual. Si bien la integración con Meta Quest 2 ha sido efectiva, futuras investigaciones podrían explorar la mejora en la interacción con el entorno, enfocándose en la fluidez y realismo de los movimientos y respuestas del usuario. Además, la integración de sensores adicionales o dispositivos de retroalimentación táctil podría enriquecer la experiencia inmersiva. También, se sugiere investigar más a fondo cómo afecta el uso de los dispositivos de realidad virtual a la SoP.

9.1.1. Trabajos Futuros

En caso de trabajo futuro se puede ampliar el área que abarcó este entorno virtual inmersivo más allá del edificio de secretaría de la carrera de Computación, además, para confirmar la fiabilidad de los resultados obtenidos, es recomendable ampliar la muestra de usuarios en los test de sensación de presencia.

10. Anexos

Anexo 1. Repositorio de código fuente del Sistema Virtual Inmersivo

<https://github.com/andresmacas/TIC>

Anexo 2. Transcripción de entrevistas técnicas para la elicitación de requisitos funcionales y no funcionales.

A continuación, se presenta la transcripción con marcas de tiempo de la entrevista realizada con los docentes tutores Andrés Roberto Navas Castellanos y Pablo Fernando Ordoñez Ordoñez.

(Contexto: se inicia el tour virtual no inmersivo realizado por el autor Pucha, con el objetivo de encontrar áreas de mejora que serán implementadas en el entorno virtual inmersivo)

(4:02) Iniciemos, por ejemplo, ya en un entorno de realidad virtual, preferiría que el menú fuera estático. (4:11) Normalmente, en los videojuegos de realidad virtual, el diseño consiste en un mundo detrás y al frente el menú inicial de energía VR. (4:20) Sí, podría ser. (4:21) Esto se denomina HUD en videojuegos.

(4:24) Por lo tanto, podría estar al frente. (4:28) Sin embargo, considerando el manejo del VR, tal vez sería más conveniente tener un mensaje inicial en lugar de una interacción que requiera pulsar "play". (4:42) Por ejemplo, que se active al presionar alguna de las punteras o simplemente al empezar a caminar. (4:45) Esto permitiría una experiencia más inmersiva desde el inicio, eliminando la necesidad de un clic adicional.

(4:55) Entonces, el inicio podría ser una vista del mundo que permanece fija mientras un mensaje en la parte inferior indique acciones como "camina con", y desde allí se active el juego. (5:08) Este enfoque se asemeja a los juegos inmersivos.

(5:27) Observemos un ejemplo. (5:34) Aquí, por ejemplo, aparece flotando una indicación sobre qué acción realizar. (5:41) Este elemento está integrado en el mundo virtual y no interrumpe la inmersión. (5:46) Sirve como tutorial inicial y posteriormente guía al usuario hacia el siguiente objetivo, como aplicar lo aprendido en una nueva sección.

(6:01) Por ejemplo, al caminar, se ofrecen instrucciones claras y concisas. (6:09) De igual manera, al aprender cómo realizar ataques, las indicaciones son prácticas. (6:13) Si el jugador desvía su atención, los elementos inmersivos atraen nuevamente su interés sin resultar invasivos. (6:27) Si el sistema detecta que el usuario está enfocado en otra actividad, los elementos pueden desaparecer temporalmente.

(6:32) En este caso, los elementos actúan como herramientas de apoyo no obligatorias, similares a un "tooltip", permitiendo que el usuario interactúe según su necesidad. (6:39) Por ejemplo, si el usuario permanece inactivo por mucho tiempo, podría indicarse mediante un mensaje sutil como "Recuerda, para moverte..." que sea accesible sin ser intrusivo.

(6:57) Asimismo, en términos de diseño, podemos trabajar con una paleta de colores definida para mantener consistencia visual. (9:03) Por ejemplo, se pueden emplear combinaciones como blanco, negro y rojo. (9:12) También sería posible personalizar la tipografía para mejorar la apariencia general del entorno.

(9:45) Otra consideración sería agregar pequeñas animaciones en secciones específicas para mejorar la experiencia del usuario. (10:05) Por ejemplo, mantener la cámara estática, pero permitiendo movimiento mediante las flechas del teclado. (10:12) Respecto a las colisiones, es fundamental corregir problemas como la superposición de elementos o colisiones inapropiadas.

(11:19) Si utilizamos fotos como parte del entorno, estas deben ajustarse adecuadamente según la perspectiva para evitar inconsistencias visuales. (12:03) Por ejemplo, una solución sería modelar ciertos elementos clave o emplear modelos de uso libre con una calidad adecuada.

(13:02) También se podría implementar una estrategia en la que las fotografías estén divididas en capas, por ejemplo, con una imagen de la vereda, otra de los arbustos delanteros y otra de los traseros, para generar mayor profundidad visual.

(13:36) Es importante considerar que los elementos en primer plano son los que más captan la atención inicial del usuario. Mejorar la calidad de estos aspectos contribuirá significativamente a la experiencia general. (13:51) En casos donde se utilicen imágenes estáticas, podría implementarse un sistema que permita alternar entre varias para simular movimiento o perspectiva dinámica.

(23:37) ...modelado. Entonces, creo que sí, ajustar ese estilo podría dar un toque más uniforme.

(23:42) Sí, sobre todo en detalles pequeños como la ropa o los accesorios que puedan llevar.

(23:45) También podríamos ver si simplificamos un poco el estilo visual para que no cargue tanto el rendimiento.

(23:48) Exactamente, porque a veces los detalles hiperrealistas pesan mucho y, al final, no aportan tanto a la experiencia general.

(23:52) ¿Qué opinan de los NPCs? ¿Agregamos más variaciones?

(23:55) Sí, creo que eso ayudaría bastante. Ahora mismo se ven muy repetitivos. Tal vez con tres o cuatro modelos diferentes ya se perciba mejor.

(23:59) De acuerdo. También podríamos randomizar un poco las animaciones de cada uno.

(24:02) Claro, y que algunos tengan acciones distintas, como caminar, sentarse o interactuar con objetos.

(24:06) Hablando de objetos, ¿qué hacemos con las texturas de los edificios? Algunas todavía parecen muy planas.

(24:10) Sí, eso hay que trabajarlo. Podríamos usar mapas normales para darles más profundidad sin necesidad de aumentar mucho la carga gráfica.

(24:14) También he visto que Blender tiene plugins para generar texturas más realistas directamente desde fotos.

(24:18) Sí, lo probé una vez y funciona bastante bien. Solo habría que asegurarnos de que las fotos sean de buena calidad.

(24:22) Otro detalle es el tema del cielo y la iluminación. Creo que falta un poco de balance ahí.

(24:26) Sí, sobre todo la iluminación. Podríamos usar luces más cálidas en ciertas áreas y frías en otras para dar contraste.

(24:30) ¿Y el cielo dinámico? He visto que algunos entornos cambian la iluminación dependiendo de la hora.

(24:34) Esa es buena idea. Podríamos hacerlo más inmersivo simulando atardeceres y noches con estrellas.

(24:38) ¿Qué hacemos con los menús y el HUD? ¿Lo dejamos minimalista o añadimos más detalles?

(24:42) Minimalista está bien, pero con algo de estilo que vaya acorde al diseño general.

(24:45) De acuerdo. Y también asegurémonos de que los textos sean legibles en cualquier resolución. Algunas fuentes pequeñas pierden calidad.

(24:49) Eso es verdad. Mejor usar fuentes claras y contrastes adecuados.

(24:53) ¿Qué más falta revisar?

(24:55) Creo que el audio y los efectos ambientales son clave para mejorar la experiencia.

(24:58) Sí, y también podríamos agregar interacción con los objetos, como abrir puertas o recoger elementos.

(25:02) Perfecto. Entonces, seguimos con esas tareas. ¿Alguna otra idea?

(25:06) Por ahora creo que eso es suficiente. Sigamos puliendo estas áreas y revisamos en la próxima reunión.

(25:10) Listo, queda anotado.

Anexo 3. Evidencia de exposición del TIC en "VR Day" en la UNACH.



Anexo 4. Repositorio de código fuente del Backend del Sistema Virtual Inmersivo

https://github.com/JeremyE7/TIC_API

Anexo 5. Cuestionario de sensación de presencia percibida

- 1. Califique su sensación de estar en el edificio administrativo de la FEIRNNR mediante el nuevo entorno inmersivo de la carrera de Computación.**

1: En ningún momento, 7: Casi todo el tiempo

- 2. ¿Hubo momentos durante su experiencia en el nuevo entorno inmersivo de la carrera de Computación en los que percibiera las situaciones como reales?**

1: En ningún momento, 7: Casi todo el tiempo

- 3. Al recordar su experiencia, ¿percibe el nuevo entorno inmersivo de la carrera de Computación más como imágenes que observó o como un lugar que realmente visitó?**

1: Imágenes que vi, 7: Algún lugar que visité

- 4. Durante su experiencia en el nuevo entorno inmersivo de la carrera de Computación, ¿fue más fuerte la sensación de estar en el edificio administrativo de la FEIRNNR o de encontrarse en otro lugar?**

1: Estar en otra parte, 7: Estar en el edificio administrativo de la FEIRNNR

- 5. Considere su recuerdo de estar en el edificio administrativo de la FEIRNNR a través del nuevo entorno inmersivo de la carrera de Computación. ¿Qué tan similar, en términos de la estructura de la memoria (memoria sensorial, memoria a corto plazo, memoria a largo plazo), percibe este entorno en comparación con cualquier otro lugar que haya visitado?**

1: Poco similar, 7: Muy similar

- 6. Durante su experiencia en el nuevo entorno inmersivo de la carrera de Computación, ¿en qué medida sintió que realmente estaba en el edificio administrativo de la FEIRNNR?**

1: No muy seguido, 7: Con frecuencia

Anexo 6. Resultado de aplicación de encuestas a 39 estudiantes

Marca temporal	Dirección de correo electrónico	Califique su sensación de estar en el edificio administrativo de la FEIRNNR mediante el nuevo entorno inmersivo de la carrera de Computación	¿Hubo momentos durante su experiencia en el nuevo entorno inmersivo de la carrera de Computación en los que percibiera las situaciones como reales?	Al recordar su experiencia, ¿percibe el nuevo entorno inmersivo de la carrera de Computación más como imágenes que observó o como un lugar que realmente visitó?	Durante su experiencia en el nuevo entorno inmersivo de la carrera de Computación, ¿fue más fuerte la sensación de estar en el edificio administrativo de la FEIRNNR o de encontrarse en otro lugar?	Considere su recuerdo de estar en el edificio administrativo de la FEIRNNR a través del nuevo entorno inmersivo de la carrera de Computación. ¿Qué tan similar, en términos de la estructura de la memoria (memoria sensorial, memoria a corto plazo, memoria a largo plazo), percibe este entorno en comparación con cualquier otro lugar	Durante su experiencia en el nuevo entorno inmersivo de la carrera de Computación, ¿en qué medida sintió que realmente estaba en el edificio administrativo de la FEIRNNR?	PROMEDIO FINAL

								que haya visitado?
27/01/2025 9:59:54	fernando.ortega@unl.edu.ec	6	4	5	4	6	5	
27/01/2025 9:59:58	luis.j.armijos@unl.edu.ec	7	7	7	7	7	7	
27/01/2025 10:00:26	jhandry.campoverde@unl.edu.ec	6	6	6	6	6	6	
27/01/2025 10:02:45	darwin.correa@unl.edu.ec	7	6	7	7	6	7	
27/01/2025 10:04:37	anderson.coello@unl.edu.ec	5	6	6	7	6	7	
27/01/2025 10:04:46	victor.roa@unl.edu.ec	7	7	4	6	7	7	
27/01/2025 10:05:21	johan.chumbi@unl.edu.ec	5	4	7	5	6	7	
27/01/2025 10:05:34	yandri.piscocama@unl.edu.ec	7	6	6	6	6	6	
27/01/2025 10:06:45	martina.maldonado@unl.edu.ec	6	4	7	5	5	5	
27/01/2025 10:09:11	jaime.landazuri@unl.edu.ec	6	5	5	6	6	7	
27/01/2025 10:11:02	ivett.zaragocin@unl.edu.ec	6	6	6	7	7	7	
27/01/2025 10:14:44	derick.vargas@unl.edu.ec	7	6	6	7	6	7	
27/01/2025 10:15:22	odalis.rosillo@unl.edu.ec	7	7	7	7	7	7	

27/01/202 5 10:16:00	maria.s.buri@unl.edu.ec	7	7	7	7	7	7	
27/01/202 5 10:16:03	mark.gonzalez@unl.edu.ec	6	5	6	6	6	6	
27/01/202 5 10:16:53	alejandro.morocho@unl.edu.ec	6	5	4	6	5	7	
27/01/202 5 10:17:26	miguel.armas@unl.edu.ec	7	7	7	7	7	7	
27/01/202 5 10:17:45	alex.sigcho@unl.edu.ec	6	6	6	7	6	6	
27/01/202 5 10:20:05	wilson.palma@unl.edu.ec	7	7	6	6	6	7	
27/01/202 5 10:52:50	luis.h.rios@unl.edu.ec	7	6	6	6	6	6	
27/01/202 5 17:22:34	yorely.romero@unl.edu.ec	6	6	7	7	6	7	
31/01/202 5 9:58:38	isaac.ramon@unl.edu.ec	6	6	7	6	7	6	
31/01/202 5 9:59:36	walter.yaguana@unl.edu.ec	7	7	7	7	6	6	
31/01/202 5 9:59:46	emilio.bravo@unl.edu.ec	7	6	6	7	7	6	
31/01/202 5 10:00:46	brian.aguinsaca@unl.edu.ec	7	7	7	7	7	7	
31/01/202 5 10:01:43	santiago.apolo@unl.edu.ec	6	5	6	6	6	5	
31/01/202 5 10:03:27	bayron.jimenez@unl.edu.ec	7	6	7	7	7	7	

31/01/2025 10:05:07	washington.apolo@unl.edu.ec	6	5	7	7	6	5	
31/01/2025 10:05:57	geovanny.romero@unl.edu.ec	6	4	5	6	6	6	
31/01/2025 10:06:55	wilman.morocho@unl.edu.ec	6	4	2	6	7	6	
31/01/2025 10:08:36	bryan.angamarca@unl.edu.ec	7	7	7	7	7	7	
31/01/2025 10:10:57	steven.luna@unl.edu.ec	6	5	5	6	5	5	
31/01/2025 10:11:40	leonardo.orozco@unl.edu.ec	6	6	6	6	7	5	
31/01/2025 10:12:40	sergio.jumbo@unl.edu.ec	6	6	5	6	6	6	
31/01/2025 10:15:42	adrian.nunez@unl.edu.ec	7	7	7	7	7	7	
31/01/2025 11:13:05	kevin.sarango@unl.edu.ec	7	7	7	6	7	7	
31/01/2025 11:34:31	luis.a.medina.c@unl.edu.ec	5	5	6	6	6	5	
31/01/2025 18:48:18	santiago.r.guachizaca@unl.edu.ec	7	7	7	7	7	7	
31/01/2025 19:31:16	luis.a.medina.c@unl.edu.ec	7	7	7	7	7	7	
Promedios		6,41025641	5,897435897	6,128205128	6,384615385	6,358974359	6,358974359	6,256410256

Anexo 7. Certificado de culminación de TIC



unl

Universidad
Nacional
de Loja

Sistema de Información Académico
Administrativo y Financiero - SIAAF

CERTIFICADO DE CULMINACIÓN Y APROBACIÓN DEL TRABAJO DE INTEGRACIÓN CURRICULAR


Nosotros, **ANDRES ROBERTO NAVAS CASTELLANOS, PABLO FERNANDO ORDOÑEZ ORDOÑEZ**, directores del Trabajo de Integración Curricular denominado **Desarrollo de realidad virtual inmersiva aplicada al entorno virtual de la carrera Computación e integración de respuestas inteligentes mediante la implementación de Inteligencia Artificial**, perteneciente al estudiante **JOSE ANDRES MACAS VELEZ**, con cédula de identidad N° **1105596918**.


Certifico:

Que luego de haber dirigido el **Trabajo de Integración Curricular**, habiendo realizado una revisión exhaustiva para prevenir y eliminar cualquier forma de plagio, garantizando la debida honestidad académica, se encuentra concluido, aprobado y está en condiciones para ser presentado ante las instancias correspondientes.

Es lo que puedo certificar en honor a la verdad, a fin de que, de así considerarlo pertinente, el/la señor/a docente de la asignatura de **Integración Curricular**, proceda al registro del mismo en el Sistema de Gestión Académico como parte de los requisitos de acreditación de la Unidad de Integración Curricular del mencionado estudiante.

Loja, 4 de Febrero de 2025


Firmado electrónicamente por:
ANDRES ROBERTO
NAVAS CASTELLANOS
F)
**DIRECTOR DE TRABAJO DE INTEGRACIÓN
CURRICULAR**


Firmado electrónicamente por:
PABLO FERNANDO
ORDOÑEZ ORDOÑEZ
F)
**DIRECTOR DE TRABAJO DE INTEGRACIÓN
CURRICULAR**


Certificado TIC/TT.: UNL-2025-000343

1/1
Educamos para Transformar

Anexo 8. Evidencia de aplicación de encuestas a estudiantes de la UNL



Figura 38.
Evidencia 1 - Exposición y aplicación de encuestas (Quinto Ciclo)



Figura 39.
Evidencia 2 - Exposición y aplicación de encuestas (Primer Ciclo)

Anexo 9. Certificado de traducción Resumen - Abstract

CERTF. N° 01.9 – 2025
Loja, 11 de marzo del 2025

El suscrito Franco Guillermo Abrigo Guarnizo.

Lcdo. En Ciencias de la Educación Mención Idioma Inglés

A petición de la parte interesada y en forma legal.

CERTIFICA:

Que **José Andrés Macas Vélez** con cédula de identidad número: **1105596918**, y **Bryan Jeremy Encalada Mejía** con cédula de identidad número: **1105844995**, estudiantes de la Carrera Ingeniería en Computación de la Facultad de la Energía las Industrias y los Recursos Naturales No Renovables de la **Universidad Nacional de Loja**, completaron satisfactoriamente la presente traducción de español a inglés del Trabajo de titulación denominado **Desarrollo de realidad virtual inmersiva aplicada al entorno virtual de la carrera Computación e integración de respuestas inteligentes mediante la implementación de Inteligencia Artificial**.

Traducción que fue guiada y revisada minuciosamente por mi persona. En consecuencia, se da validez a la presentación de la misma. Es todo cuanto puedo certificar en honor a la verdad, pudiendo los estudiantes hacer uso del presente documento en lo que estimaren conveniente.

Atentamente,



.....
Franco Guillermo Abrigo Guarnizo

Lcdo. En Ciencias de la Educación Mención Idioma Inglés

Número de Registro Senescyt: 1008-2021-2368808

Cédula: 1104492127

email: franco.abrigo@hotmail.com

celular: 0990447198

11. Bibliografía

- [1] J. L. Pucha Banegas, “Realidad Virtual aplicada a la información del proceso académico estudiantil de la Carrera de Computación de la Facultad de la Energía, las Industrias, y los Recursos Naturales no Renovables de la Universidad Nacional de Loja”, bachelorThesis, Universidad Nacional de Loja, 2023. Consultado: el 7 de octubre de 2024. [En línea]. Disponible en: <https://dspace.unl.edu.ec//handle/123456789/28435>
- [2] J. J. Cummings y J. N. Bailenson, “How Immersive Is Enough? A Meta-Analysis of the Effect of Immersive Technology on User Presence”, *Media Psychol.*, vol. 19, núm. 2, pp. 272–309, abr. 2016, doi: 10.1080/15213269.2015.1015740.
- [3] J. Xiong, E.-L. Hsiang, Z. He, T. Zhan, y S.-T. Wu, “Augmented reality and virtual reality displays: emerging technologies and future perspectives”, *Light Sci. Appl.*, vol. 10, núm. 1, p. 216, oct. 2021, doi: 10.1038/s41377-021-00658-8.
- [4] S. Martirosov, M. Bureš, y T. Zítka, “Cyber sickness in low-immersive, semi-immersive, and fully immersive virtual reality”, *Virtual Real.*, vol. 26, núm. 1, pp. 15–32, mar. 2022, doi: 10.1007/s10055-021-00507-4.
- [5] B. N. Peterson *et al.*, “Immersive Virtual Reality: A Safe, Scalable, Non-opioid Analgesic for Military and Veteran Patients”, *Front. Virtual Real.*, vol. 2, nov. 2021, doi: 10.3389/frvir.2021.742290.
- [6] N. Li, N. Sun, C. Cao, S. Hou, y Y. Gong, “Review on visualization technology in simulation training system for major natural disasters”, *Nat. Hazards*, vol. 112, núm. 3, pp. 1851–1882, jul. 2022, doi: 10.1007/s11069-022-05277-z.
- [7] “A Literature Review of Using Machine Learning in Software Development Life Cycle Stages | IEEE Journals & Magazine | IEEE Xplore”. Consultado: el 4 de febrero de 2025. [En línea]. Disponible en: <https://ieeexplore.ieee.org/abstract/document/9568959>
- [8] “EXTREME PROGRAMMING VS SCRUM: A COMPARISON OF AGILE MODELS | International Journal of Technology Innovation and Management (IJTIM)”. Consultado: el 4 de febrero de 2025. [En línea]. Disponible en: <https://www.journals.gaftim.com/index.php/ijtim/article/view/77>
- [9] J. Polcar, M. Gregor, P. Horejsi, y P. Kopecek, “Methodology for Designing Virtual Reality Applications”, en *DAAAM Proceedings*, 1a ed., vol. 1, B. Katalinic, Ed., DAAAM International Vienna, 2016, pp. 0768–0774. doi: 10.2507/26th.daaam.proceedings.107.
- [10] “Classifying Presence Scores: Insights and Analysis from Two Decades of the Igroup Presence Questionnaire (IPQ) | ACM Transactions on Computer-Human Interaction”. Consultado: el 3 de febrero de 2025. [En línea]. Disponible en: <https://dl.acm.org/doi/abs/10.1145/3689046>
- [11] O. Nannipieri, “Do Presence Questionnaires Actually Measure Presence? A Content Analysis of Presence Measurement Scales”, en *Extended Reality*, L. T. De Paolis, P. Arpaia, y M. Sacco, Eds., Cham: Springer International Publishing, 2022, pp. 273–295. doi: 10.1007/978-3-031-15546-8_24.
- [12] A. T. Jebb, V. Ng, y L. Tay, “A Review of Key Likert Scale Development Advances: 1995–2019”, *Front. Psychol.*, vol. 12, may 2021, doi: 10.3389/fpsyg.2021.637547.
- [13] J. Robinson, “Likert Scale”, en *Encyclopedia of Quality of Life and Well-Being Research*, F. Maggino, Ed., Cham: Springer International Publishing, 2023, pp. 3917–3918. doi: 10.1007/978-3-031-17299-1_1654.
- [14] R. Toro, M. Peña-Sarmiento, B. L. Avendaño-Prieto, S. Mejía-Vélez, y A. Bernal-Torres, “Análisis Empírico del Coeficiente Alfa de Cronbach según Opciones de Respuesta, Muestra y Observaciones Atípicas”, *Rev. Iberoam. Diagnóstico Eval. - E Aval. Psicológica*, vol. 2, núm. 63, p. 17, 2022.
- [15] “¿Qué tan apropiadamente reportaron los autores el Coeficiente del Alfa de Cronbach?” Consultado: el 3 de febrero de 2025. [En línea]. Disponible en: <https://cathi.uacj.mx/handle/20.500.11961/18789>
- [16] M. Slater, D. Banakou, A. Beacco, J. Gallego, F. Macia-Varela, y R. Oliva, “A Separate Reality: An Update on Place Illusion and Plausibility in Virtual Reality”, *Front. Virtual Real.*, vol. 3, jun. 2022, doi: 10.3389/frvir.2022.914392.

- [17] “The role of vision and proprioception in self-motion encoding: An immersive virtual reality study | Attention, Perception, & Psychophysics”. Consultado: el 3 de febrero de 2025. [En línea]. Disponible en: <https://link.springer.com/article/10.3758/s13414-021-02344-8>
- [18] “User experiences of virtual reality technologies for healthcare in learning: an integrative review”. Consultado: el 21 de enero de 2025. [En línea]. Disponible en: <https://www.tandfonline.com/doi/epdf/10.1080/0144929X.2020.1788162?needAccess=true>
- [19] M. Melo *et al.*, “Immersive multisensory virtual reality technologies for virtual tourism”, *Multimed. Syst.*, vol. 28, núm. 3, pp. 1027–1037, jun. 2022, doi: 10.1007/s00530-022-00898-7.
- [20] J.-H. Li y B.-C. Lee, “Efficient Topology Study for Auto-Generated Animation - Focusing on Mixamo -”, *만화애니메이션 연구*, pp. 31–52, jun. 2023, doi: 10.7230/KOSCAS.2023.71.031.
- [21] J. Wang *et al.*, “Zero-shot Pose Transfer for Unrigged Stylized 3D Characters”, en *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada: IEEE, jun. 2023, pp. 8704–8714. doi: 10.1109/CVPR52729.2023.00841.
- [22] J. J. Miranda Trujillo y K. F. Santillan Segura, “Diseño de un agente virtual basado en metahumanos para establecer puntos de consulta de procesos internos de la universidad”, 2023, Consultado: el 3 de febrero de 2025. [En línea]. Disponible en: <http://www.dspace.espol.edu.ec/handle/123456789/60573>
- [23] “C# 11 | SpringerLink”. Consultado: el 3 de febrero de 2025. [En línea]. Disponible en: https://link.springer.com/chapter/10.1007/978-1-4842-8619-7_26
- [24] A. Troelsen y P. Japikse, “Introducing C# and .NET 6”, en *Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming*, A. Troelsen y P. Japikse, Eds., Berkeley, CA: Apress, 2022, pp. 3–26. doi: 10.1007/978-1-4842-7869-7_1.
- [25] “Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine”, Unity. Consultado: el 21 de enero de 2025. [En línea]. Disponible en: <https://unity.com>
- [26] J. Fitzgerald, T. Oda, y H. D. Macedo, “Proceedings of the 18th International Overture Workshop”, el 19 de enero de 2021, *arXiv*: arXiv:2101.07261. doi: 10.48550/arXiv.2101.07261.
- [27] “La herramienta de creación 3D en tiempo real más potente - Unreal Engine”. Consultado: el 21 de enero de 2025. [En línea]. Disponible en: <https://www.unrealengine.com/es-ES>
- [28] “Godot Engine - Free and open source 2D and 3D game engine”. Consultado: el 21 de enero de 2025. [En línea]. Disponible en: <https://godotengine.org/>
- [29] B. Foundation, “blender.org - Home of the Blender project - Free and Open 3D Creation Software”, blender.org. Consultado: el 21 de enero de 2025. [En línea]. Disponible en: <https://www.blender.org/>
- [30] “Autodesk Maya | Get Prices & Buy Official Maya 2025 | Autodesk”. Consultado: el 21 de enero de 2025. [En línea]. Disponible en: <https://www.autodesk.com/products/maya/overview>
- [31] “Autodesk 3ds Max Software | Get Prices & Buy Official 3ds Max 2025”. Consultado: el 21 de enero de 2025. [En línea]. Disponible en: <https://www.autodesk.com/products/3ds-max/overview>
- [32] C. Osborne, J. Ding, y H. R. Kirk, “The AI community building the future? A quantitative analysis of development activity on Hugging Face Hub”, *J. Comput. Soc. Sci.*, vol. 7, núm. 2, pp. 2067–2105, oct. 2024, doi: 10.1007/s42001-024-00300-8.
- [33] B. Gupta, P. Mittal, y T. Mufti, “A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services”, presentado en Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India, mar. 2021.

- Consultado: el 3 de febrero de 2025. [En línea]. Disponible en:
<https://eudl.eu/doi/10.4108/eai.27-2-2020.2303255>
- [34] A. Moniz, M. Gordon, I. Bergum, M. Chang, y G. Grant, "Speech Services", en *Beginning Azure Cognitive Services: Data-Driven Decision Making Through Artificial Intelligence*, A. Moniz, M. Gordon, I. Bergum, M. Chang, y G. Grant, Eds., Berkeley, CA: Apress, 2021, pp. 153–192. doi: 10.1007/978-1-4842-7176-6_5.
- [35] S. G. P. Ibarra, J. R. Quispe, F. F. Mullicundo, y D. A. Lamas, "HERRAMIENTAS Y TECNOLOGÍAS PARA EL DESARROLLO WEB DESDE EL FRONTEND AL BACKEND", 2021.
- [36] R. Vyas, "Comparative Analysis on Front-End Frameworks for Web Applications", *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, núm. 7, pp. 298–307, jul. 2022, doi: 10.22214/ijraset.2022.45260.
- [37] I. H. Madurapperuma, M. S. Shafana, y M. J. A. Sabani, "State-of-Art Frameworks for Front-end and Back-end Web Development", 2022.
- [38] K. Saundariya, M. Abirami, K. R. Senthil, D. Prabakaran, B. Srimathi, y G. Nagarajan, "Webapp Service for Booking Handyman Using MongoDB, Express JS, React JS, Node JS", en *2021 3rd International Conference on Signal Processing and Communication (ICSPC)*, Coimbatore, India: IEEE, may 2021, pp. 180–183. doi: 10.1109/ICSPC51351.2021.9451783.
- [39] C. L. Vidal-Silva *et al.*, "Experiencia académica en desarrollo rápido de sistemas de información web con Python y Django", *Form. Univ.*, vol. 14, núm. 5, pp. 85–94, oct. 2021, doi: 10.4067/S0718-50062021000500085.
- [40] S. R. Modugu y H. Farhat, "Implementation of the Internet of Things Application Based on Spring Boot Microservices and REST Architecture", en *Software Engineering Perspectives in Intelligent Systems*, vol. 1294, R. Silhavy, P. Silhavy, y Z. Prokopova, Eds., en *Advances in Intelligent Systems and Computing*, vol. 1294. , Cham: Springer International Publishing, 2020, pp. 20–31. doi: 10.1007/978-3-030-63322-6_3.
- [41] "Node.js — Download Node.js®". Consultado: el 3 de febrero de 2025. [En línea]. Disponible en: <https://nodejs.org/en/download>
- [42] "Django", Django Project. Consultado: el 3 de febrero de 2025. [En línea]. Disponible en: <https://www.djangoproject.com/>
- [43] "Spring Boot :: Spring Boot". Consultado: el 3 de febrero de 2025. [En línea]. Disponible en: <https://docs.spring.io/spring-boot/index.html>
- [44] "Meta Quest 2 Touch Controllers", Inspirit. Consultado: el 3 de febrero de 2025. [En línea]. Disponible en: <https://help.inspiritvr.com/hc/en-us/articles/15488182886804-Meta-Quest-2-Touch-Controllers>
- [45] Saran. R, Shrikesh. S.P, Vamsidharan. V, y Sangeetha. V, "Virtual Reality Based Moon And Space Station", en *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, Coimbatore, India: IEEE, may 2021, pp. 481–484. doi: 10.1109/ICSPC51351.2021.9451691.
- [46] D. E. Rzig, N. Iqbal, I. Attisano, X. Qin, y F. Hassan, "Virtual Reality (VR) Automated Testing in the Wild: A Case Study on Unity-Based VR Applications", en *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, en ISSTA 2023. New York, NY, USA: Association for Computing Machinery, jul. 2023, pp. 1269–1281. doi: 10.1145/3597926.3598134.
- [47] J. M. González-Perellón, R. Durón-Figueroa, S. Bouchard, y G. Cárdenas-López, "Evaluación de usabilidad, deseo de jugar y sentido de presencia en ambientes virtuales para el tratamiento del juego compulsivo", *Psicol. Iberoam.*, vol. 29, núm. 1, jun. 2021, doi: 10.48102/pi.v29i1.302.
- [48] L. B. Cadet y H. Chainay, "Memory of virtual experiences: Role of immersion, emotion and sense of presence", *Int. J. Hum.-Comput. Stud.*, vol. 144, p. 102506, dic. 2020, doi: 10.1016/j.ijhcs.2020.102506.
- [49] T. F. Lønne, H. R. Karlsen, E. Langvik, y I. Saksvik-Lehouillier, "The effect of immersion on sense of presence and affect when experiencing an educational scenario in virtual

reality: A randomized controlled study”, *Heliyon*, vol. 9, núm. 6, jun. 2023, doi: 10.1016/j.heliyon.2023.e17196.