

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE SOFTWARE

Tema:

SISTEMA DE REALIDAD VIRTUAL DEDICADO A LA IDENTIFICACIÓN Y ARMADO DEL INSTRUMENTAL QUIRÚRGICO PARA ESTUDIANTES DE ENFERMERÍA

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero de Software

ÁREA: Desarrollo de software

LÍNEA DE INVESTIGACIÓN: Software, Tecnologías de la información y Ciencia de datos

AUTORES:
Anderson Daniel Sánchez Sandoval
Kevin Alexander Saquinga Chango
TUTOR: Ing. Oscar Fernando Ibarra Torres, Mg.

Ambato - Ecuador
julio – 2025

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA DE REALIDAD VIRTUAL DEDICADO A LA IDENTIFICACIÓN Y ARMADO DEL INSTRUMENTAL QUIRÚRGICO PARA ESTUDIANTES DE ENFERMERÍA , desarrollado bajo la modalidad Proyecto de Investigación por los señores Anderson Daniel Sánchez Sandoval y Kevin Alexander Saquinga Chango, estudiantes de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que los estudiantes han sido tutorados durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, julio 2025.

Firmado electrónicamente por:
OSCAR FERNANDO IBARRA TORRES
Validar únicamente con FirmaEC

Ing. Oscar Fernando Ibarra Torres, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA DE REALIDAD VIRTUAL DEDICADO A LA IDENTIFICACIÓN Y ARMADO DEL INSTRUMENTAL QUIRÚRGICO PARA ESTUDIANTES DE ENFERMERÍA es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, julio 2025.

Anderson Daniel Sánchez Sandoval
C.C. 1805192794
AUTOR

Kevin Alexander Saquinga Chango
C.C. 1850747013
AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato , julio 2025.

Anderson Daniel Sánchez Sandoval
C.C. 1805192794
AUTOR

Kevin Alexander Saquinga Chango
C.C. 1850747013
AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por los señores Anderson Daniel Sánchez Sandoval y Kevin Alexander Saquinga Chango estudiantes de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE REALIDAD VIRTUAL DEDICADO A LA IDENTIFICACIÓN Y ARMADO DEL INSTRUMENTAL QUIRÚRGICO PARA ESTUDIANTES DE ENFERMERÍA, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con el señor Presidente del Tribunal.

Ambato, julio 2025.

Firmado electrónicamente por:
FRANKLIN OSWALDO MAYORGA MAYORGA
Validar únicamente con FirmaEC

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.
PRESIDENTE DEL TRIBUNAL

Firmado electrónicamente por:
FRANKLIN OSWALDO MAYORGA MAYORGA
Validar únicamente con FirmaEC

Ing. Franklin Mayorga Mayorga, Mg.
PROFESOR CALIFICADOR

Firmado electrónicamente por:
DAVID OMAR GUEVARA AULESTIA
Validar únicamente con FirmaEC

Ing. David Guevara Aulestia, Mg.
PROFESOR CALIFICADOR

DEDICATORIA – DANIEL SÁNCHEZ

Esta tesis está dedicada a todas las personas que han sido parte esencial de mi vida, aquellas que, con su presencia, palabras, enseñanzas y afecto, han contribuido de manera invaluable a la persona que soy hoy. Cada uno, desde su lugar, ha dejado una huella imborrable en este camino de formación y crecimiento.

Dedico este trabajo, en primer lugar, a mi madre, cuyo amor incondicional, esfuerzo constante y ejemplo de fortaleza han sido pilares fundamentales en cada etapa de mi vida. Su apoyo ha sido la guía silenciosa que me ha impulsado a seguir adelante incluso en los momentos más difíciles.

A mi familia, por estar presente con palabras de aliento, paciencia y comprensión, brindándome el soporte emocional necesario para alcanzar esta meta.

A mí mismo, por no rendirme ante los obstáculos, por creer en mis capacidades y por haber recorrido con determinación el camino que me ha llevado hasta este logro.

DEDICATORIA – KEVIN SAQUINGA

Esta tesis está dedicada a todas las personas que han sido parte esencial de mi vida y de este proceso académico, contribuyendo con su apoyo, amor y ejemplo a que hoy pueda alcanzar esta meta.

A Dios, por brindarme la fortaleza en los momentos de incertidumbre, por iluminar mi camino con sabiduría y por estar presente en cada paso de este recorrido.

A mis padres, por su amor incondicional, por enseñarme con su ejemplo el valor del esfuerzo, la honestidad y la perseverancia. Gracias por estar siempre, en la distancia o en silencio, sosteniéndome sin condiciones y confiando en mi capacidad incluso cuando yo mismo dudaba.

A mi familia, por cada palabra de aliento, por su paciencia y comprensión durante los días de desvelo, y por ser mi refugio emocional en cada etapa de este proceso.

A mí mismo, por la constancia, por no rendirme cuando el camino se hizo difícil, por cada hora de dedicación, y por demostrarme que los sueños se alcanzan con compromiso y determinación.

AGRADECIMIENTO – DANIEL SANCHEZ

Agradezco profundamente a todas las personas e instituciones que formaron parte fundamental de este proceso y contribuyeron, de una u otra forma, a la culminación de esta tesis.

A mi madre, por ser mi mayor inspiración y fortaleza. Su amor incondicional, sus sacrificios y su constante apoyo emocional han sido la base sobre la cual he construido cada uno de mis logros. Su ejemplo de entrega y perseverancia me ha guiado en cada paso de este camino.

Expreso mi sincero agradecimiento a mi tutor, al ***Ing. Oscar Fernando Ibarra Torres*** *por su orientación, paciencia y valiosas observaciones durante el desarrollo de este trabajo. Su guía fue esencial para mantener el enfoque y la calidad del proyecto.*

A los profesionales que generosamente compartieron su conocimiento y experiencia: al ***Dr. José Luis Herrera*** *y a la* ***Dra. Marianela Mejías de Duarte****, por sus valiosos aportes técnicos y académicos, que enriquecieron significativamente el contenido de esta tesis.*

A la ***Universidad Técnica de Ambato****, por brindarme el espacio académico, las herramientas y la formación que me han permitido crecer como profesional y como ser humano. Esta institución ha sido el escenario donde he forjado mi vocación y visión profesional.*

Finalmente, extiendo mi agradecimiento a mi compañero de desarrollo ***Kevin Saquinga****, por su compromiso, colaboración y entrega durante la ejecución de este proyecto. Su apoyo técnico y humano fue esencial para alcanzar los objetivos propuestos.*

AGRADECIMIENTO – KEVIN SAQUINGA

*En primer lugar, agradezco profundamente a **Dios**, por ser mi guía espiritual y por darme la fortaleza para completar esta etapa académica. Culminar esta tesis representa no solo el cierre de una etapa académica, sino también el resultado del esfuerzo conjunto de muchas personas a quienes deseo expresar mi sincero agradecimiento.*

*A mi tutor, al **Ing. Oscar Fernando Ibarra Torres**, por su guía constante, su paciencia y su compromiso con el desarrollo de este proyecto. Su orientación fue clave para que esta propuesta de simulación en realidad virtual tome forma y calidad.*

*A los profesionales que colaboraron con sus conocimientos y experiencia durante el desarrollo del sistema: **Dr. José Luis Herrera** y **Dra. Marianela Mejías de Duarte**.*

Quienes contribuyeron de manera significativa en la validación del sistema, aportando una visión práctica y profesional que enriqueció el proyecto.

*A la **Universidad Técnica de Ambato**, a la **Facultad de Ingeniería en Sistemas, Electrónica e Industrial** y la **Facultad de Ciencias de la Salud**, por proporcionar el entorno académico y los recursos necesarios para la investigación y ejecución de este trabajo.*

*A mi compañero de desarrollo, **Daniel Sánchez**, por su compromiso, esfuerzo compartido y trabajo en equipo, sin el cual este proyecto no hubiera sido posible. Y a todas aquellas personas que, de manera directa o indirecta, han sido parte de este proceso, mi más profundo agradecimiento.*

ÍNDICE GENERAL DE CONTENIDOS

PORTADA ..... **i**

APROBACIÓN DEL TUTOR ..... **ii**

AUTORÍA ..... **iii**

DERECHOS DE AUTOR ..... **iv**

APROBACIÓN DEL TRIBUNAL DE GRADO ..... **v**

DEDICATORIA – DANIEL SÁNCHEZ ..... **vi**

DEDICATORIA – KEVIN SAQUINGA ..... **vii**

AGRADECIMIENTO – DANIEL SANCHEZ ..... **viii**

AGRADECIMIENTO – KEVIN SAQUINGA ..... **ix**

ÍNDICE GENERAL DE CONTENIDOS ..... **x**

ÍNDICE DE TABLAS ..... **xiv**

ÍNDICE DE FIGURAS ..... **xx**

ÍNDICE DE ANEXOS ..... **xxxi**

RESUMEN EJECUTIVO ..... **xxxii**

ASTRACT ..... **xxxiii**

CAPÍTULO I. MARCO TEÓRICO **1**

1.1 Tema de investigación 1

1.1.1 Planteamiento del problema 1

1.2 Antecedentes investigativos 2

1.3 Fundamentación teórica 15

1.3.1 Realidad Virtual en la Educación Médica 15

1.3.2 Simulación de Emergencias y Entrenamiento en Habilidades Clínicas 16

1.3.3 Ventajas de la Realidad Virtual frente a Métodos Tradicionales 16

1.3.4 Desafíos y Limitaciones en el Uso de Realidad Virtual en Educación Médica 16

1.3.5 La realidad virtual y la realidad aumentada 17

1.3.6 Funcionamiento del mundo virtual 17

1.3.7 La realidad virtual en el marco de la enseñanza médica 18

1.3.8 Motores para desarrollo de VR 18

1.3.9 Lenguajes de Programación 20

1.3.10 Herramientas de Modelado 3D 21

1.3.11 Complejidad en el Desarrollo de Entornos de Realidad Virtual para Educación en Enfermería 22

1.3.12 Métricas para medir complejidad de desarrollo 24

1.3.13 Métricas de Rendimiento en Tiempo Real 24

1.3.14 Métricas de Experiencia de Usuario (UX) 24

1.3.15 Metodologías de desarrollo 25

1.3.16 Metodologías Tradicionales 25

1.3.17 Metodologías Ágiles 26

1.4 Objetivos 28

1.4.1 Objetivo general 28

1.4.2 Objetivos específicos 28

CAPÍTULO II. METODOLOGÍA 29

2.1 Materiales 29

2.2 Métodos 30

2.2.1 Modalidad de la investigación 30

2.2.2 Población y muestra 31

2.2.3 Recolección de información 32

2.2.4 Procesamiento y análisis de datos 33

CAPÍTULO III. RESULTADOS Y DISCUSIÓN 50

3.1 Tablas comparativas de Motores, Lenguajes, Gafas de RV y Herramientas de Modelado 50

3.2 Evaluación Complejidad herramientas de desarrollo de VR 57

3.3 Análisis de requerimientos 65

3.3.1 Requerimientos Funcionales 65

3.3.2 Requerimientos no Funcionales 66

3.3.3 Especificación de Requerimientos de Software (SRS) basado en la norma ISO 29148:2018 66

3.4 Métrica Cyclomatic Complexity (complejidad de desarrollo) 73

3.4.1 Métrica Halstead Complexity (complejidad de desarrollo) 80

3.5 FPS Frames Per Second (Rendimiento en Tiempo Real) 88

3.5.1 Uso de CPU/GPU (Rendimiento en Tiempo Real) 96

3.6 Tiempo de Respuesta a Inputs (Experiencia de Usuario (UX)) 106

3.7 Índice de Usabilidad (Experiencia de Usuario (UX)) 136

3.8 Metodología Extreme Programming (XP) 141

3.8.1 Fases de Extreme Programming (XP) 143

3.9 Metodología planteada para el Desarrollo de Realidad Virtual 198

3.9.1 Fase de Investigación y Análisis 199

4.1.1 Fase de Diseño 200

4.1.2 Fase de Desarrollo 202

4.1.3 Fase de Pruebas 207

4.1.4 Fase de Implementación 209

4.1.5 Fase de Mantenimiento y Mejora Continua 210

4.2 Validación de la metodología 228

4.2.1 Design Science Research (DSR) – Investigación Basada en Diseño 229

4.2.2 Etapas de Design Science Research 229

4.2.3 Resultados de la encuesta TAM 236

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES 241

5.1 Conclusiones 241

5.2 Recomendaciones 242

Referencias Bibliográficas 243

ÍNDICE DE TABLAS

Tabla 1. Motores de desarrollo VR 51

Tabla 2. Lenguajes de programación comunes en VR 52

Tabla 3. Comparación de Gafas de realidad virtual 52

Tabla 4. Herramientas de modelado 3D 53

Tabla 5: Tabla Comparativa de Herramientas de Realidad Virtual en Cirugía 55

Tabla 6: Tecnologías Utilizadas por los estudios 56

Tabla 7: Análisis de Desempeño Técnico de Unity 3D 60

Tabla 8: Evaluación de la Complejidad Técnica del Motor Unreal Engine 60

Tabla 9: Evaluación de la Complejidad Técnica del Software Blender 61

Tabla 10: Evaluación de la Complejidad Técnica del Software Autodesk Maya 62

Tabla 11: Distribución Comparativa del Índice de Complejidad (IC) 63

Tabla 12: Características de los usuarios a usar el sistema 69

Tabla 13: Requerimientos No Funcionales 71

Tabla 14: Resumen general de funcionalidades y uso de una herramienta de análisis de calidad de código 74

Tabla 15: Resumen de Visual Studio Analyzer: características, ventajas y aplicación técnica 75

Tabla 16: Mapa Comparativo SonarQube vs Visual Studio Analyzer 75

Tabla 17: Resumen de Understand: funciones, ventajas y enfoque técnico 81

Tabla 18: Resumen de Cppcheck: análisis, beneficios y aplicación técnica 82

Tabla 19: Mapa Comparativo Understand vs Cppcheck 83

Tabla 20: Métrica Halstead Complexity del script AsignarInstrumento.cs 83

Tabla 21: Métrica Halstead Complexity del script AsignarInstrumentoGeneral.cs .. 84

Tabla 22: Métrica Halstead Complexity del script BotonExpandir.cs 84

Tabla 23: Métrica Halstead Complexity del script BotonInstrumento.cs 84

Tabla 24: Métrica Halstead Complexity del script BotonInstrumentoGeneral.cs 85

Tabla 25: Métrica Halstead Complexity del script CambiarMenus.cs 85

Tabla 26: Métrica Halstead Complexity del script CanvasController.cs 85

Tabla 27: Métrica Halstead Complexity del script CargaAsincrona.cs 85

Tabla 28: Métrica Halstead Complexity del script ControladorExpandir.cs 86

Tabla 29: Métrica Halstead Complexity del script ControlMesaMayo.cs 86

Tabla 30: Métrica Halstead Complexity del script ControlTiemposGeneral.cs 86

Tabla 31: Métrica Halstead Complexity del script LoadSceneWithMessage.cs 86

Tabla 32: Métrica Halstead Complexity del script MenuPrincipalUta.cs 87

Tabla 33: Métrica Halstead Complexity del script MostrarNombreObjetoVR.cs 87

Tabla 34: Métrica Halstead Complexity del script MultiplayerCreationControllerGeneral.cs 87

Tabla 35: Métrica Halstead Complexity del script ShowObjectNameOnHover.cs ... 87

Tabla 36: Métrica Halstead Complexity del script UIInteractionController.cs 88

Tabla 37: Resumen de Unity Profiler: monitoreo de rendimiento en entornos Unity 89

Tabla 38: Resumen de OVR Metrics Tool: monitoreo de rendimiento en dispositivos Meta Quest 90

Tabla 39: Resumen de Unreal Insights: análisis avanzado de rendimiento en Unreal Engine 91

Tabla 40: Mapa comparativo Unity Profile, Herramienta de métricas OVR y Unreal Insights 92

Tabla 41: Resumen de NVIDIA Nsight: análisis gráfico avanzado para entornos de alto rendimiento 97

Tabla 42: Características y usos de RenderDoc: herramienta de depuración gráfica multiplataforma 98

Tabla 43: Mapa comparativo NVIDIA Nsight vs RenderDoc 99

Tabla 44: Tiempo de respuesta a Inputs de la primera semana de prueba del sistema 107

Tabla 45: Tiempo de respuesta a Inputs de la cuarta semana de prueba del sistema 108

Tabla 46: Información técnica y funcional de NASA-TLX para medir carga mental en sistemas interactivos 137

Tabla 47: Información técnica y funcional de Loop11 138

Tabla 48: Mapa comparativo Nasa-TLX vs Loop 11 139

Tabla 49: Promedios obtenidos por los estudiantes en cada dimensión del método NASA-TLX 140

Tabla 50: Historia de Usuario: Navegar por el menú principal 146

Tabla 51: Historia de Usuario: Navegar por el menú del tutorial 147

Tabla 52: Historia de Usuario: Navegar por el escenario de simulación quirúrgico 147

Tabla 53: Historia de Usuario: Selección de operación 148

Tabla 54: Historia de Usuario: Selección de tiempos quirúrgicos 148

Tabla 55: Historia de Usuario: Selección de los instrumentos quirúrgicos 148

Tabla 56: Historia de Usuario: Selección de los instrumentos quirúrgicos 149

Tabla 57: Historia de Usuario: Selección de los instrumentos quirúrgicos 149

Tabla 58: Historia de Usuario: Mostrar tiempo total 150

Tabla 59: Historia de Usuario: Mensaje de finalización 150

Tabla 60: Historia de Usuario: Reiniciar el sistema al volver 150

Tabla 61: Historia de Usuario: Interacción XR con objetos 151

Tabla 62: Historia de Usuario: Aleatorización de instrumentos 151

Tabla 63: Tarjeta CRC de MenuPrincipal 156

Tabla 64: Tarjeta CRC de CargaAsincrona 156

Tabla 65: Tarjeta CRC de MultiplayerCreationControllerGeneral 156

Tabla 66: Tarjeta CRC de ControlTiemposGeneral 157

Tabla 67: Tarjeta CRC de AsignarInstrumentoGeneral 157

Tabla 68: Tarjeta CRC de BotonInstrumentoGeneral 157

Tabla 69: Tarjeta CRC de ToggleMesa 157

Tabla 70: Tarjeta CRC de MensajeErrores 158

Tabla 71: Tarjeta CRC de MenuOperacion 158

Tabla 72: Tarjeta CRC de PanelInstrumentos 158

Tabla 73: Tarjeta CRC de ControlMesaMayo 158

Tabla 74: Tarjeta CRC de ControladorExpandir 159

Tabla 75: Tarjeta de Ingeniería de MenuPrincipal 159

Tabla 76: Tarjeta de Ingeniería de CargaAsincrona 159

Tabla 77: Tarjeta de Ingeniería de MenuOperacion 160

Tabla 78: Tarjeta de Ingeniería de MenuInstrumentalGeneral 160

Tabla 79: Tarjeta de Ingeniería de ControlTiemposGeneral..... 160

Tabla 80: Tarjeta de Ingeniería de AsignarInstrumentoGeneral..... 160

Tabla 81: Tarjeta de Ingeniería de ToggleMesa 160

Tabla 82: Tarjeta de Ingeniería de PanelMesaMayo..... 161

Tabla 83: Tarjeta de Ingeniería de PanelErrores..... 161

Tabla 84: Tarjeta de Ingeniería de BotonInstrumentoGeneral..... 161

Tabla 85:Tarjeta de Ingeniería de PanelInstrumentos..... 161

Tabla 86: Tarjeta de Ingeniería de ControlMesaMayo 161

Tabla 87: Tarjeta de Ingeniería de ControladorExpandir 162

Tabla 88: Tarjeta de Ingeniería de Aleatorización del ScrollView..... 162

Tabla 89: Caso de Prueba de aceptación: Navegación en el entorno virtual 189

Tabla 90: Caso de Prueba de aceptación: Navegación en el entorno virtual 189

Tabla 91: Caso de Prueba de aceptación: Exploración del entorno quirúrgico 190

Tabla 92: Caso de Prueba de aceptación: Selección del tipo de operación quirúrgica 190

Tabla 93: Caso de Prueba de aceptación: Elección del tiempo quirúrgico..... 191

Tabla 94: Caso de Prueba de aceptación: Asignación de instrumentos quirúrgicos 191

Tabla 95: Caso de Prueba de aceptación: Activación de la mesa Mayo..... 191

Tabla 96: Caso de Prueba de aceptación: Sistema de advertencias por errores 192

Tabla 97: Caso de Prueba de aceptación: Visualización del tiempo total de armado 192

Tabla 98: Caso de Prueba de aceptación: Mensaje de finalización de la actividad . 193

Tabla 99: Caso de Prueba de aceptación: Reinicio completo del escenario quirúrgico 193

Tabla 100: Caso de Prueba de aceptación: Aleatorización del orden de los instrumentos 193

Tabla 101: Caso de Prueba de aceptación: Identificación de objetos con puntero láser 194

Tabla 102: Etapas que existen Design Science Research (DSR) – Investigación Basada en Diseño 229

Tabla 103: Actividades a ejecutar para evaluar la metodología 236

Tabla 104: Promedio y Media de la Utilidad Percibida de la Metodología 237

Tabla 105: Promedio y Media de la Facilidad de Uso Percibida de la Metodología 238

Tabla 106: Promedio y Media de la Actitud hacia el Uso de la Metodología 239

Tabla 107: Promedio y Media de la Intención de Uso de la Metodología 240

ÍNDICE DE FIGURAS

Figura 1:Cálculo del Alfa de Cronbach de resultados obtenidos 34

Figura 2: Frecuencia de uso de herramientas tecnológicas 35

Figura 3: Uso de Realidad Virtual para Fines Educativos 35

Figura 4: Percepción de la Utilidad de la Realidad Virtual en la Identificación de Instrumentos Quirúrgicos 36

Figura 5: Nivel de Satisfacción con las Herramientas Tecnológicas 37

Figura 6: Frecuencia de Uso: Aprendizaje en Línea vs. Presencial 37

Figura 7: Herramientas Tecnológicas Más Utilizadas en el Aprendizaje 38

Figura 8: Conocimiento Previo sobre Realidad Virtual 38

Figura 9: Opiniones sobre la Eficacia de la Realidad Virtual en la Educación 39

Figura 10: Barreras Percibidas para el Uso de Realidad Virtual en su formación académica 40

Figura 11: Interés en Capacitación sobre Tecnologías Emergentes 40

Figura 12: Frecuencia de Participación en Capacitación Tecnológica 41

Figura 13: Efecto de la Pandemia en el Uso de Herramientas Tecnológicas 42

Figura 14: Preferencias de Formato de Capacitación (Virtual, Presencial, Híbrido). 42

Figura 15: Impacto de la Realidad Virtual en las Habilidades Prácticas 43

Figura 16: El Futuro del Aprendizaje y las Tecnologías Emergentes 44

Figura 17:Evidencia del Análisis en ATLAS.ti de la Entrevista al Experto en Desarrollo 45

Figura 18:Evidencia del Análisis en ATLAS.ti de la Entrevista de la Licenciada en Enfermería 46

Figura 19:Nube de Palabras de la Entrevista al Experto en Desarrollo 47

Figura 20:Nube de Palabras de la Entrevista a la Licenciada en Enfermería 47

Figura 21:Representación de la Estructuras Conceptuales: Entrevista al Experto en Desarrollo 49

Figura 22:Representación de la Estructuras Conceptuales: Entrevista a la Licenciada en Enfermería 49

Figura 23: Análisis de Calidad de Código Dashboard de SonarQube 76

Figura 24:Análisis de Calidad de Código del Directorio Scripts - Métricas SonarQube 77

Figura 25: Análisis de Vulnerabilidad de Seguridad 78

Figura 26:Problemas de confiabilidad detectados por SonarQube 78

Figura 27: Problemas de mantenibilidad detectados por SonarQube debido al uso del atributo obsoleto 79

Figura 28: Revisión de Security Hotspots en SonarQube 79

Figura 29: FPS en Menú Inicial 93

Figura 30: FPS en observando todos los objetos del quirófano 94

Figura 31: FPS sin muchos objetos 94

Figura 32: FPS con muy pocos objetos 95

Figura 33: Resultados de Monitoreo de Rendimiento en Tiempo Real con NVIDIA Nsight Graphics 100

Figura 34:Monitoreo de Rendimiento en Simulación de Quirófano Virtual 100

Figura 35: Supervisión de Recursos en una Simulación XR Médica 101

Figura 36:Análisis de Rendimiento en una Simulación Médica 3D 102

Figura 37: Carga de GPU y CPU con múltiples colas de espera 103

Figura 38: Bloqueo de hilos por colas de GPU en procesos del sistema 104

Figura 39:Procesos del sistema con colas críticas de GPU 105

Figura 40: Análisis de Desempeño del Sistema mediante Nvidia Systems 105

Figura 41: Frecuencia de resultados: pregunta 1, semana 1 109

Figura 42: Frecuencia de resultados: pregunta 1, semana 2 110

Figura 43: Frecuencia de resultados: pregunta 1, semana 3 111

Figura 44: Frecuencia de resultados: pregunta 1, semana 4 111

Figura 45: Frecuencia de resultados: pregunta 2, semana 1 112

Figura 46: Frecuencia de resultados: pregunta 2, semana 2 113

Figura 47: Frecuencia de resultados: pregunta 2, semana 3 114

Figura 48: Frecuencia de resultados: pregunta 2, semana 4 114

Figura 49: Frecuencia de resultados: pregunta 3, semana 1 115

Figura 50: Frecuencia de resultados: pregunta 3, semana 2 116

Figura 51: Frecuencia de resultados: pregunta 3, semana 3 116

Figura 52: Frecuencia de resultados: pregunta 3, semana 4 117

Figura 53: Frecuencia de resultados: pregunta 4, semana 1 118

Figura 54: Frecuencia de resultados: pregunta 4, semana 2 118

Figura 55: Frecuencia de resultados: pregunta 4, semana 3 119

Figura 56: Frecuencia de resultados: pregunta 4, semana 4..... 120

Figura 57: Frecuencia de resultados: pregunta 5, semana 1..... 120

Figura 58: Frecuencia de resultados: pregunta 5, semana 2..... 121

Figura 59: Frecuencia de resultados: pregunta 5, semana 3..... 122

Figura 60: Frecuencia de resultados: pregunta 5, semana 4..... 122

Figura 61: Frecuencia de resultados: pregunta 6, semana 1..... 123

Figura 62: Frecuencia de resultados: pregunta 6, semana 2..... 124

Figura 63: Frecuencia de resultados: pregunta 6, semana 3..... 124

Figura 64: Frecuencia de resultados: pregunta 6, semana 4..... 125

Figura 65: Frecuencia de resultados: pregunta 7, semana 1..... 126

Figura 66: Frecuencia de resultados: pregunta 7, semana 2..... 126

Figura 67: Frecuencia de resultados: pregunta 7, semana 3..... 127

Figura 68: Frecuencia de resultados: pregunta 7, semana 4..... 127

Figura 69: Frecuencia de resultados: pregunta 8, semana 1..... 128

Figura 70: Frecuencia de resultados: pregunta 8, semana 2..... 129

Figura 71: Frecuencia de resultados: pregunta 8, semana 3..... 129

Figura 72: Frecuencia de resultados: pregunta 8, semana 4..... 130

Figura 73: Frecuencia de resultados: pregunta 9, semana 1..... 131

Figura 74: Frecuencia de resultados: pregunta 9, semana 2..... 131

Figura 75: Frecuencia de resultados: pregunta 9, semana 3..... 132

Figura 76: Frecuencia de resultados: pregunta 9, semana 4..... 132

Figura 77: Frecuencia de resultados: pregunta 10, semana 1 133

Figura 78: Frecuencia de resultados: pregunta 10, semana 2 134

Figura 79: Frecuencia de resultados: pregunta 10, semana 3 134

Figura 80: Frecuencia de resultados: pregunta 10, semana 4 135

Figura 81: Diagrama de casos de uso 152

Figura 82: Diagrama de casos de uso: Flujo completo de navegación entre menús 153

Figura 83: Diagrama de casos de uso: Simulación de Operación General 154

Figura 84: Diagrama de casos de uso: Seleccionar Herramienta Correcta/Incorrecta 154

Figura 85: Menú Principal del sistema NurseTools 163

Figura 86: Tutorial de cómo usar el movimiento de los joysticks 163

Figura 87: Tutorial de cómo activar los punteros 164

Figura 88: Tutorial de como interactuar dentro del escenario del quirófano 164

Figura 89: Tutorial de cómo salir del sistema 165

Figura 90: Escenario del Quirófano 165

Figura 91: Menú Nurse Tools que te permite seleccionar el tipo de operación y tiempo 166

Figura 92: Menú del instrumental quirúrgico 167

Figura 93: Canvas para ver información extra de la imagen 167

Figura 94: Canva de la mesa Mayo 168

Figura 95: Canva para el control de errores y tiempo total 168

Figura 96: Diagrama de actividad: Inicio y Posicionamiento en el Escenario 169

Figura 97: Diagrama de actividad: Presionar Botón "Iniciar" y Abrir Menú de Operación 170

Figura 98: Diagrama de actividad: Selección de Tipo y Tiempo de Operación 170

Figura 99: Diagrama de actividad: Selección de Instrumentos 171

Figura 100: Diagrama de actividad: Consulta de Información de Instrumentos 171

Figura 101: Diagrama de actividad: Simulación Completa de la Operación 172

Figura 102: Diagrama de secuencia: Inicio de la Aplicación y Posicionamiento en el Escenario 173

Figura 103: Diagrama de secuencia: Presionar Botón "Iniciar" 173

Figura 104: Diagrama de secuencia: Seleccionar Tipo y Tiempo de Operación 174

Figura 105: Diagrama de secuencia: Navegar al Menú de Selección de Herramientas 174

Figura 106: Diagrama de secuencia: Selección Correcta de Herramientas 175

Figura 107: Diagrama de secuencia: Selección Incorrecta de Herramientas 175

Figura 108: Diagrama de secuencia: Expansión de Imagen de Herramienta y Consulta de Información 176

Figura 109: Diagrama de secuencia: Finalizar Simulación General 177

Figura 110: Script CanvasController para la activación de los canvas 178

Figura 111: Script CargaAsincrona para el cambio de escenarios 179

Figura 112: Script MostrarNombreObjetoVR para ver el nombre de los objetos con los punteros 180

Figura 113: Script MenuPrincipalUta para controlar los canvas de la primera escena 181

Figura 114: Script UIInteractionController para los controladores de manos 182

Figura 115: Script ControladorExpandir para la visualización del instrumental quirúrgico 183

Figura 116: Commit creación del escenario del quirófano 186

Figura 117: Commit de creación del menú principal y el Canva de tutorial 186

Figura 118:Commit de arreglo de la pantalla congelada al ingresar al sistema..... 187

Figura 119: Commit creación del Canva Expansión de imagen 187

Figura 120: Commit creación de los canvas del menú de operación y del menú del instrumental quirúrgico 187

Figura 121: Metodología para el Desarrollo de Aplicaciones de Realidad Virtual (RV) para Escenarios de Enfermería en Cirugía 198

Figura 122: Commit de la versión beta Funcional 203

Figura 123: Escenario del quirófano diseñado para su versión beta 204

Figura 124: Menú de entrenamiento para la identificación del instrumental quirúrgico 204

Figura 125: Módulo principal la interacción del instrumental quirúrgico 205

Figura 126: Módulo de entrenamiento del sistema 205

Figura 127: Módulo de retroalimentación del instrumental..... 206

Figura 128: Módulo de evaluación dependiendo del tipo de operación que seleccione 206

Figura 129: Módulo de resultados una vez terminado la evaluación..... 207

Figura 130: Primer commit desarrollado correspondiente a la creación del diseño del escenario..... 211

Figura 131:Diseño del quirófano correspondiente al primer commit 211

Figura 132: Segundo commit desarrollado correspondiente a la creación del menú Principal y el menú del instrumental quirúrgico 212

Figura 133: Menú principal correspondiente al segundo commit..... 212

Figura 134: Menú del instrumental quirúrgico correspondiente al segundo commit 212

Figura 135: Tercer commit correspondiente a la creación del Canva de expansión para ver información del instrumental seleccionado..... 213

Figura 136: Canva de expansión y detalle de las características principales de la herramienta seleccionada 213

Figura 137: Cuarto commit correspondiente a la creación del Canva de expansión para ver información del instrumental seleccionado..... 213

Figura 138: Asignación de todos los botones para dar click aparezca el instrumento en el Canva de expansión 214

Figura 139: Quinto commit correspondiente a la configuración de la cámara del avatar 214

Figura 140: Punto de vista de la cámara a partir del avatar 215

Figura 141: Sexto commit correspondiente a la configuración del avatar..... 215

Figura 142: Configuración del avatar para que tenga animaciones 215

Figura 143:Séptimo commit correspondiente al Canva que simula la mesa mayo.. 216

Figura 144: Mesa mayo con sus respectivos tiempos quirúrgicos..... 216

Figura 145: Octavo commit correspondiente a cambios de la mesa mayo 216

Figura 146: Mesa mayo junto al menú del instrumental quirúrgico 217

Figura 147: Noveno commit correspondiente a la creación de un aviso si se comete errores 217

Figura 148: Mensaje de error al seleccionar mal un instrumento 217

Figura 149: Décimo commit correspondiente a la configuración de los botones para la interacción con la mesa mayo 218

Figura 150: Asignación del instrumental quirúrgico en la mesa mayo 218

Figura 151: Onceavo commit correspondiente al Toggle para activar la mesa mayo 218

Figura 152: Configuración del Toggle al dar click se activa el Canva de la mesa mayo 219

Figura 153: Configuración del Toggle al dar volver a click se desactiva el Canva de la mesa mayo 219

Figura 154: Doceavo commit correspondiente al control de la cámara del avatar .. 219

Figura 155: Visión de la cámara integrada con el avatar 220

Figura 156: Treceavo commit correspondiente a los cambios de los muebles del escenario 220

Figura 157: Control para la activación de los objetos (mesa semi luna y de la mesa mayo) 221

Figura 158: Catorceavo commit correspondiente a los canvas que nos permiten evaluar el armado de la mesa mayo 221

Figura 159: Menú principal más el botón General funcional 222

Figura 160: Menú de los instrumentos para la ver el aprendizaje 222

Figura 161: Quinceavo commit centrado en la creación de los nombres flotantes .. 223

Figura 162: Creación de nombre flotante al pasar los punteros 223

Figura 163: Décimo sexto commit enfocado en la creación del menú de bienvenida con sus tutoriales 223

Figura 164: Menú de bienvenida al sistema de realidad virtual..... 224

Figura 165: Tutorial del uso de los joisntink 224

Figura 166: Décimo séptimo commit enfocado en la asignación del resto de los nombres flotantes a los objetos 224

Figura 167: Nombres flotantes asignados al resto de los objetos 225

Figura 168: Décimo octavo commit centrado en la corrección de la pantalla congelada..... 225

Figura 169: Canva de Cargando … para controlar la pantalla congelada..... 225

Figura 170: Décimo noveno commit centrado en el Canva que te permite ver los resultados del armado de la mesa mayo 226

Figura 171: Canva para ver los errores cometidos en el armado 226

Figura 172: Ultimo commit desarrollado para el sistema 226

Figura 173: Canva de notificación de finalizar todos los tiempos 227

Figura 174: Menú de tipo de operación corregido 227

Figura 175: Menú de tiempo de operación corregido 228

Figura 176: Capacitación del armado de la mesa mayo..... 231

Figura 177: Capacitación de un quirófano funcional en el HGDA..... 232

Figura 178: Capacitación de los instrumentos quirúrgicos 232

Figura 179: Demostración práctica del entorno de realidad virtual 234

Figura 180: Práctica guiada en la navegación del entorno simulado 234

Figura 181: Instrucciones para la interacción dentro del simulador de realidad virtual. 235

Figura 182: Capacitación en la manipulación de instrumentos virtuales en la mesa mayo 235

Figura 183: Reunión entre desarrolladores para discutir avances del sistema. 235

ÍNDICE DE ANEXOS

Anexo 1: Encuesta 257

Anexo 2: Entrevistas 262

Anexo 3: Encuesta de Usabilidad 264

Anexo 4: Cuestionario del método NASA TLX (Task Load Index) 266

Anexo 5: Metodo de evaluación TAM 267

Anexo 6: Manual de instalación del sistema 269

Anexo 7: Manual de Usuario 274

RESUMEN EJECUTIVO

Las limitaciones en la infraestructura de los laboratorios clínicos dificultan la formación práctica en instrumentación quirúrgica para estudiantes de Enfermería. Como respuesta, se desarrolló un sistema de realidad virtual que simula entornos quirúrgicos abiertos, permitiendo identificar instrumentos y practicar el armado de la mesa Mayo en procedimientos como cesáreas y apendicetomías.

Se logró una experiencia inmersiva en entornos quirúrgicos simulados mediante una aplicación desarrollada en Unity con C# y ejecutada en gafas Meta Quest 3. El sistema fue diseñado con un enfoque iterativo y centrado en el usuario, aplicando la metodología Extreme Programming (XP). Para evaluar la complejidad y eficiencia del desarrollo, se creó una metodología propia que integró métricas como la Complejidad Ciclomática, Complejidad de Halstead, rendimiento en tiempo real y evaluación de la experiencia del usuario a través de los instrumentos NASA-TLX y SUS. La validación de esta metodología se realizó bajo el enfoque Design Science Research, lo que permitió estructurar científicamente el proceso de evaluación.

Los resultados reflejan que el 93 % de los estudiantes calificó como útil o muy útil el sistema para identificar instrumental, y el 86,2 % indicó que facilita la práctica fuera del laboratorio. Además, el cuestionario TAM aplicado a 8 expertos arrojó promedios de 4.8 en utilidad percibida, 4.45 en facilidad de uso, 4.67 en actitud hacia el uso y 4.75 en intención de adopción en escala de Likert 5.

En futuras fases, incorporara un sistema de gestión de usuarios con control de asistencias y reanudación del progreso.

Palabras clave: Realidad virtual, instrumentación quirúrgica, gafas meta quest 3, Unity, C#, Extreme Programming, métricas de complejidad, usabilidad.

ASTRACT

Limitations in clinical laboratory infrastructure hinder practical training in surgical instrumentation for nursing students. In response, a virtual reality system was developed that simulates open surgical environments, allowing students to identify instruments and practice setting up the Mayo table in procedures such as cesarean sections and appendectomies.

An immersive experience in simulated surgical environments was achieved through an application developed in Unity with C# and run on Meta Quest 3 glasses. The system was designed with an iterative and user-centered approach, applying the Extreme Programming (XP) methodology. To evaluate the complexity and efficiency of the development, a proprietary methodology was created that integrated metrics such as Cyclomatic Complexity, Halstead Complexity, real-time performance, and user experience evaluation through the NASA-TLX and SUS instruments. The validation of this methodology was carried out under the Design Science Research approach, which allowed the evaluation process to be scientifically structured.

The results show that 93% of students rated the system as useful or very useful for identifying instruments, and 86.2% indicated that it facilitates practice outside the laboratory. In addition, the TAM questionnaire applied to eight experts yielded averages of 4.8 in perceived usefulness, 4.45 in ease of use, 4.67 in attitude toward use, and 4.75 in intention to adopt on a 5-point Likert scale.

In future phases, it will incorporate a user management system with attendance control and progress resumption.

Keywords: Virtual reality, surgical instrumentation, Meta Quest 3 glasses, Unity, C#, Extreme Programming, complexity metrics, usability.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

SISTEMA DE REALIDAD VIRTUAL DEDICADO A LA IDENTIFICACIÓN Y ARMADO DEL INSTRUMENTAL QUIRÚRGICO PARA ESTUDIANTES DE ENFERMERÍA

1.1.1 Planteamiento del problema

Históricamente, la educación médica ha sido una combinación de conocimientos teóricos y exposición práctica. Los métodos tradicionales de enseñanza médica, aunque han sido probados en el tiempo, presentan desafíos financieros, éticos y logísticos que, en el panorama médico actual, acelerado y en constante evolución, a menudo resultan limitantes [1]. En la actualidad, la Universidad Técnica de Ambato enfrenta un problema de sobrecarga en sus instalaciones de práctica clínica, especialmente en el área de identificación y armado del instrumento antes de una cirugía. Las instalaciones físicas disponibles solo pueden albergar un promedio de 17 estudiantes, mientras que más de 100 estudiantes están matriculados en materias las cuales necesitan del uso de estas instalaciones, lo que limita el acceso a prácticas clínicas esenciales. Esta situación no solo impide que los estudiantes reciban una formación adecuada, sino que también compromete la calidad educativa en el área de la atención de emergencias médicas [2].

La búsqueda de nuevas tecnologías como la realidad virtual (RV) podría solucionar este problema al crear un entorno virtual que simula una sala de primeros auxilios. Estudios previos han demostrado que la RV no solo aumenta la capacidad de formación, sino que también mejora significativamente la inmersión y el aprendizaje de los estudiantes. El uso de simulaciones realistas en RV ha permitido que más del 90% de los 591 participantes experimenten mejoras en su habilidad para manejar situaciones de emergencia, superando las barreras que las simulaciones físicas tradicionales no pueden sortear [2], [3]. Además, esta tecnología ofrece un espacio

seguro para que los estudiantes puedan practicar múltiples veces, enfrentándose a una amplia gama de escenarios sin los riesgos asociados a la práctica real [3], [4].

Por ello el entrenamiento práctico de técnicas médicas es fundamental en el proceso de enseñanza de los estudiantes de la FCS. Sin embargo, en la Universidad Técnica de Ambato, los estudiantes enfrentan el reto de contar con un solo laboratorio que tiene una capacidad máxima de 12 personas. Por tanto, la falta de infraestructura adecuada se presenta como un problema crítico, pero puede ser mitigado a través de la adopción de tecnologías avanzadas como la RV. Por cada ciclo académico normalmente existen 4 cursos los cual. Esto obliga a que se practiquen rotaciones semanales, lo que limita la enseñanza directa y continua de las actividades necesarias para su formación. Esta situación afecta la experiencia formativa y retrasa el desarrollo de sus habilidades. Para abordar este desafío, la universidad busca integrar tecnologías innovadoras, como la simulación en RV para la identificación y el correcto orden de los instrumentos quirúrgicos, que permitirá a los estudiantes practicar de manera segura y efectiva en un entorno simulado. Sin embargo, a pesar de sus beneficios, la implementación de la VR en la educación médica también enfrenta ciertos desafíos, como los costos iniciales de los equipos y la curva de aprendizaje para los usuarios.

1.2 Antecedentes investigativos

En el artículo [5], se describe el desarrollo de una aplicación de realidad virtual enfocada en la enseñanza de anatomía y práctica médica. La simulación se lleva a cabo en un entorno de sala de emergencias y utiliza el dispositivo Meta Quest 2, junto con el Google VR SDK, para ofrecer una experiencia inmersiva. La plataforma de desarrollo incluye Android Studio, proporcionando gráficos de baja calidad, pero con fidelidad visual media. La retroalimentación se limita a estímulos visuales, sin incluir háptica ni sonora, lo cual restringe la percepción sensorial durante la simulación. Para evaluar el aprendizaje de los estudiantes, se implementan pruebas de desempeño. Entre los desafíos técnicos se identifican dificultades en la representación realista de las partes del cuerpo. Además, el sistema presenta restricciones en cuanto a su uso, aunque es reproducible para fines educativos.

La investigación [6], se centra en el uso de la realidad virtual para la formación en cirugía abierta en un entorno de quirófano. Aunque no se especifica el dispositivo de realidad virtual utilizado, la simulación ofrece gráficos de calidad media y alta fidelidad visual, proporcionando una experiencia inmersiva para el entrenamiento de procedimientos invasivos. La retroalimentación se limita a estímulos visuales, sin incluir háptica o sonora, lo que podría influir en la percepción completa de la simulación. Para evaluar la efectividad del aprendizaje, se utilizan pruebas de desempeño que permiten medir el progreso de los usuarios. Los desafíos técnicos identificados incluyen la adaptación de la simulación a la realidad virtual y la gestión de los recursos limitados del sistema. A pesar de estas limitaciones, el simulador es reproducible, lo que facilita su uso en programas de formación médica.

En este contexto, en la investigación [7], se muestra una revisión de literatura que explora nuevas metodologías para la simulación de intervenciones quirúrgicas abiertas en realidad virtual, con un enfoque en la formación y educación médica. La simulación se ambienta en un entorno de quirófano, aunque no se especifica el dispositivo de realidad virtual utilizado debido a la naturaleza teórica del artículo. Se discuten aspectos como la calidad gráfica y la fidelidad visual, pero no se proporciona una evaluación detallada de estos elementos, ya que el artículo se centra más en revisar enfoques existentes. La retroalimentación visual se menciona como una característica importante para el entrenamiento en realidad virtual. Se destacan desafíos como la adaptación de las técnicas quirúrgicas a un entorno virtual y la necesidad de estándares reproducibles en la educación médica mediante simulación.

Así mismo, el artículo [8], es una revisión de la literatura que aborda las prácticas actuales, desafíos y oportunidades en la creación de aplicaciones de realidad aumentada y virtual. Aunque no se centra en un tipo específico de simulación, el artículo proporciona una visión general de cómo estas tecnologías se aplican en diversos contextos, como la simulación médica en entornos de sala de emergencias o quirófano. La discusión no se limita a una plataforma especifica de desarrollo, aunque se menciona Unity como una de las herramientas populares en el campo. La revisión destaca la importancia de la retroalimentación visual y la necesidad de prácticas reproducibles para asegurar la calidad en el entrenamiento. Además, aborda desafíos técnicos, como la integración efectiva de las tecnologías en diferentes aplicaciones y

la falta de estándares unificados. La literatura también identifica oportunidades para mejorar la educación y la formación mediante estas tecnologías emergentes.

De acuerdo con la investigación presentada por [9] se enfoca en el diseño y la evaluación de una interfaz inmersiva en realidad virtual para entrenar a los profesionales de la salud en habilidades clínicas. La simulación tiene lugar en un entorno de sala de emergencias, con gráficos de alta calidad que proporcionan una experiencia visual realista y detallada. Aunque no se especifica el dispositivo de realidad virtual utilizado, la plataforma se desarrolla con Unity para maximizar la inmersión y la efectividad del entrenamiento. La retroalimentación se centra en estímulos visuales, sin incluir háptica o sonora, lo cual limita la variedad de la percepción sensorial. Los desafíos técnicos identificados incluyen la adaptación de la interfaz a la realidad virtual y la gestión de recursos limitados en el sistema. La metodología utilizada en el desarrollo de la interfaz asegura su reproducibilidad, facilitando la implementación en programas de formación clínica.

En el trabajo de [10] se centra en el desarrollo de un sistema de simulación en tiempo real para la educación en anestesiología utilizando realidad virtual. El entorno simula un quirófano y utiliza el software Unity, aunque no se especifica el dispositivo de realidad virtual utilizado. Tanto la calidad gráfica como la fidelidad visual se califican como medias, proporcionando una experiencia visual suficiente pero no excepcional. El sistema ofrece retroalimentación visual y pruebas de desempeñó, pero carece de retroalimentación háptica o sonora. Los desafíos técnicos incluyen dificultades en la representación realista de los escenarios, mientras que las limitaciones reportadas se refieren al realismo de la retroalimentación. El sistema es reproducible y permite la verificación de los resultados mediante estudios futuros.

El artículo [11], describe el diseño y desarrollo de una aplicación de aprendizaje a distancia basada en realidad virtual para la enseñanza de medicina manual y terapia. La simulación se desarrolla en un entorno de sala de emergencias y utiliza el software Unity, aunque no se especifica el dispositivo de realidad virtual utilizado. La calidad gráfica se clasifica como alta, mientras que la fidelidad visual es media. La simulación ofrece retroalimentación visual y pruebas de desempeñó, pero no incluye retroalimentación háptica o sonora. Entre los desafíos técnicos reportados está la falta

de experiencia previa en el diseñó de este tipo de aplicaciones. Aunque no se reportan limitaciones importantes, el sistema es reproducible, permitiendo su implementación en otros contextos educativos.

En la investigación [12] se explora el valor de la realidad aumentada y la realidad virtual en la formación de cirujanos. Aunque no se especifica un dispositivo de realidad virtual particular, el enfoque es una revisión de la literatura existente sobre simuladores en cirugía. El software Unity es mencionado como herramienta clave, pero no se ofrecen detalles sobre la calidad gráfica o la fidelidad visual. La simulación se centra en la retroalimentación visual y en pruebas de desempeño, junto con autoevaluaciones para los usuarios. El artículo también aborda los desafíos y limitaciones, como la falta de retroalimentación sonora y háptica, y plantea dudas sobre la relación costo-beneficio de estos simuladores. Es una revisión reproducible y refutable, lo que permite la verificación de sus conclusiones en estudios futuros.

Según [13] aborda el diseño de un simulador sintético para la enseñanza de habilidades quirúrgicas abiertas, enfocándose en la exploración de extremidades en situaciones de trauma. La simulación se realiza en un entorno de quirófano, aunque no se especifica el uso de dispositivos de realidad virtual o software adicional. La calidad gráfica es media, pero la fidelidad visual es alta, lo que garantiza una representación adecuada de los procedimientos quirúrgicos. El simulador ofrece retroalimentación visual y pruebas de desempeño, aunque no se incluye retroalimentación háptica o sonora. Los desafíos técnicos más destacados incluyen la complejidad en la creación de entornos virtuales detallados para la simulación. El sistema es reproducible, permitiendo su uso en diferentes contextos de formación, y es refutable, abriendo la puerta a futuras mejoras y evaluaciones.

La propuesta de [14] presenta una aplicación de realidad virtual enfocada en la simulación basada en la inercia, con énfasis en la interacción en entornos de sala de emergencias. Aunque no se especifica el dispositivo de realidad virtual utilizado, la simulación destaca por su alta calidad gráfica y fidelidad visual, lo que ofrece una experiencia inmersiva y realista. La retroalimentación visual es un componente clave, así como las pruebas de desempeño y la autoevaluación para los usuarios. Un desafío técnico importante es la precisión en la simulación de la interacción robótica, mientras

que la limitación principal reportada es el realismo en la retroalimentación. El sistema es reproducible y refutable, lo que permite futuras evaluaciones y mejoras.

De acuerdo con [15], examina las tendencias actuales del uso de la realidad virtual en la educación médica, centrándose en su implementación en entornos de sala de emergencias. Aunque no se especifica un dispositivo de realidad virtual particular, la revisión aborda principalmente el software Unity como una herramienta clave en el desarrollo de estas aplicaciones. Se destaca la retroalimentación visual y las pruebas de desempeño, aunque no se incluyen elementos hápticos o sonoros. La revisión también señala la importancia de la autoevaluación en los entornos de simulación médica. Entre los desafíos técnicos, se menciona la necesidad de mejorar la retroalimentación visual y sonora para crear simulaciones más realistas. El artículo es reproducible y refutable, permitiendo que sus conclusiones sean verificadas y actualizadas a través de futuras investigaciones.

En la investigación [16], se realizó un estudio sobre el uso de la realidad virtual en cirugía invasiva, utilizando cámaras Meta Quest y auriculares para simular procedimientos. La tecnología de visión por computador permitió captar movimientos precisos, mejorando la experiencia de aprendizaje. La población del estudio consistió en 22 estudiantes de medicina, quienes interactuaron con la interfaz UX diseñada para optimizar el entrenamiento quirúrgico. Los resultados indicaron que la inmersión en el entorno virtual facilitó la adquisición de habilidades técnicas, proporcionando una alternativa efectiva a los métodos tradicionales.

Según [17] menciona que la arquitectura de un simulador de entrenamiento inmersivo para cirugía invasiva suele estar compuesta por varios componentes clave que trabajan en conjunto para ofrecer una experiencia realista y efectiva. A continuación, se describen los elementos principales de dicha arquitectura:

- Sistema de Realidad Virtual (RV).
- Captura de Movimientos y Sensores Hápticos.
- Interfaz de Usuario (UX) Los movimientos hápticos replican la sensación táctil de las herramientas quirúrgicas y el tejido humano.

Gracias a la inmersión total y a una interfaz intuitiva, los usuarios pueden practicar técnicas complejas con precisión y recibir retroalimentación en tiempo real sobre su desempeño, lo que mejora tanto la destreza como la confianza en un entorno seguro y controlado. Esta arquitectura combina hardware avanzado y software especializado para ofrecer una plataforma robusta y eficaz en la formación de cirujanos.

Como señala [18] la simulación de cirugía mediante dispositivos como el Meta Quest 2, que ofrecen gráficos detallados y alta fidelidad visual, está transformando la educación médica al permitir a los cirujanos practicar procedimientos complejos en un entorno seguro y controlado. El uso de RV con alta resolución visual facilita el desarrollo de habilidades técnicas al reproducir de forma precisa la anatomía y las herramientas quirúrgicas, lo cual es crucial para lograr una experiencia inmersiva y educativa. No obstante, la implementación de estos simuladores enfrenta desafíos técnicos mixtos. Aunque el realismo gráfico es notable, la falta de retroalimentación táctil avanzada sigue limitando la representación de sensaciones físicas críticas durante el procedimiento quirúrgico, como la resistencia de los tejidos. Para evaluar la eficacia de estas simulaciones, se utilizan tanto evaluaciones objetivas (como el tiempo de respuesta y la precisión en tareas específicas) como evaluaciones subjetivas (mediante cuestionarios y observaciones sobre la experiencia del usuario). Este enfoque de evaluación permite obtener una visión integral de los beneficios y limitaciones del simulador, capturando no solo el rendimiento técnico del usuario, sino también la percepción de realismo e inmersión. Desde un método científico con enfoque mixto, que combina métodos cuantitativos y cualitativos, los estudios sobre simuladores quirúrgicos de RV buscan proporcionar datos sólidos sobre la transferencia de habilidades al entorno clínico real. Este enfoque permite adaptar continuamente la tecnología a las necesidades y desafíos de la práctica médica, promoviendo una formación quirúrgica más eficaz y alineada con los estándares de seguridad y calidad en la atención al paciente.

En la investigación de [19] SYNAPSE VINCENT, es una plataforma innovadora que integra tecnología de endoscopia virtual tridimensional para la formación y entrenamiento en procedimientos quirúrgicos. Esta herramienta permite a los médicos y estudiantes de medicina practicar en un entorno simulado que replica de manera precisa los desafíos que enfrentarían en una sala de operaciones. Dentro de este

contexto, LapSim y TeamSim son sistemas complementarios que ofrecen simulaciones específicas para cirugía laparoscópica y entrenamiento en equipo, respectivamente. LapSim proporciona un entorno para perfeccionar habilidades quirúrgicas mínimamente invasivas, mientras que TeamSim permite a los equipos de cirugía practicar la comunicación y colaboración necesarias durante procedimientos complejos. También reseña que el uso de dispositivos como el Oculus Rift potencia aún más la experiencia de inmersión, permitiendo a los usuarios interactuar con el entorno virtual de manera intuitiva y realista. Esta combinación de tecnologías no solo mejora la competencia técnica, sino que también promueve la confianza y la capacidad de trabajo en equipo entre los profesionales de la salud.

En la investigación [20] se menciona que la simulación quirúrgica mediante plataformas integradas está transformando la formación médica, especialmente para estudiantes de medicina de ambos sexos menores de 35 años. Estas plataformas utilizan simuladores de RV uretral con gráficos detallados y entornos quirúrgicos realistas, permitiendo a los futuros cirujanos practicar técnicas avanzadas en escenarios que replican fielmente la anatomía y las condiciones clínicas. Con evaluaciones estructuradas y una metodología científica rigurosa, estos simuladores ofrecen una retroalimentación precisa sobre el desempeño del estudiante, ayudando a mejorar sus habilidades quirúrgicas. Esta tecnología no solo refuerza el aprendizaje, sino que también prepara a los estudiantes para enfrentar procedimientos reales con mayor confianza y competencia.

El estudio realizado por [21] demuestra que las simulaciones de alta fidelidad que combinan realidad virtual (RV) y aumentada están diseñadas para imitar intervenciones quirúrgicas reales con un nivel de realismo excepcional. La calidad gráfica y la fidelidad visual son cruciales para recrear con precisión los detalles anatómicos y los entornos quirúrgicos, mientras que los sistemas de retroalimentación háptica permiten sentir las variaciones de resistencia de los tejidos, lo que simula la interacción física en procedimientos quirúrgicos. Sin embargo, el desafío radica en lograr una simulación precisa de esas variaciones y resistencias que se encuentran en la realidad, lo que destaca la necesidad de mejoras continuas en la tecnología. Junto a la telemedicina ha revolucionado varios aspectos de la atención médica, y su impacto en la capacitación y educación quirúrgica ha sido particularmente significativo. Al

aprovechar la tecnología para cerrar las brechas geográficas, la telemedicina brinda a los aprendices de cirugía acceso a orientación experta, retroalimentación en tiempo real y experiencias de aprendizaje inmersivas, independientemente de su ubicación. Estas innovaciones son clave para perfeccionar las habilidades de los cirujanos en formación, asegurando mayor seguridad y éxito en las operaciones reales.

El artículo [22] la simulación de cirugía abierta ha ganado relevancia, con la educación médica como una herramienta eficaz para entrenar a cirujanos en un entorno controlado y seguro. Una revisión literaria sobre este tema revela que las simulaciones permiten a los estudiantes practicar procedimientos complejos, como incisiones y manipulaciones de órganos, sin poner en riesgo a pacientes reales. Estas simulaciones suelen incluir modelos anatómicos de alta fidelidad y retroalimentación háptica para replicar las sensaciones físicas de una cirugía abierta. La literatura relacionada con el tema destaca también el impacto positivo de estas tecnologías en la mejora de las habilidades técnicas, la reducción del tiempo de aprendizaje y el aumento de la confianza del cirujano. Sin embargo, algunos estudios subrayan la necesidad de estandarizar los protocolos de simulación y de desarrollar metodologías que incluyan tanto aspectos técnicos como no técnicos, como la toma de decisiones en tiempo real. En conjunto, la simulación de cirugía abierta es vista como una herramienta fundamental para la capacitación quirúrgica, aunque se requiere investigación continua para optimizar su eficacia y accesibilidad.

La propuesta de [23], plantea que la simulación de realidad virtual se ha convertido en una herramienta poderosa para la enseñanza de habilidades quirúrgicas, brindando a los estudiantes y profesionales un escenario seguro y controlado en el que practicar. En estos entornos simulados, los participantes reciben retroalimentación tanto visual como háptica, lo cual permite una experiencia de aprendizaje más inmersiva. Sin embargo, uno de los desafíos críticos es la falta de modelos suficientemente realistas que puedan imitar con precisión la textura y respuesta de los tejidos humanos, dificultando la reproducción exacta de las condiciones de un quirófano real. También existe una necesidad de simuladores en tiempo real es cada vez más evidente, especialmente aquellos que puedan integrar datos detallados sobre las propiedades físicas de los órganos vivos. Esta demanda impulsa el desarrollo de sistemas de simulación que sean más sofisticados y fieles a la realidad, permitiendo una práctica

más efectiva y orientada a las condiciones reales de los procedimientos médicos. Este estudio se llevó a cabo bajo un enfoque descriptivo cualitativo, evaluando las experiencias de una población de 22 individuos, compuesta por 11 educadores en técnicas quirúrgicas y 11 aprendices (trainees) en entrenamiento. Este enfoque permitió comprender las limitaciones y las oportunidades de la simulación en el desarrollo de habilidades quirúrgicas, subrayando la importancia de incorporar simuladores más precisos y detallados en los programas de formación médica.

En el estudio [24], se genera una revisión literaria sobre la educación médica asistida por simuladores, como el TurpMentor, resalta su importancia en la formación de cirujanos al proporcionar un entorno seguro para la práctica sin poner en riesgo a los pacientes. En el contexto de la prostatectomía robótica, los simuladores ofrecen un recurso valioso para mejorar habilidades quirúrgicas en modelos 3D, enfocándose en la precisión y el rendimiento en procedimientos complejos. El TurpMentor, específicamente diseñado para entrenamiento en resección transuretral de próstata (TURP), permite capturar datos en tiempo real y evaluar el desempeño de los usuarios mediante métricas cuantitativas y cualitativas. Si bien el simulador tiene como objetivo optimizar la calidad y la seguridad de la atención al paciente, algunos estudios indican limitaciones en el realismo, como la ausencia de simulación precisa del sangrado, lo que puede afectar la inmersión y la transferencia de habilidades a situaciones reales. Estos aspectos limitan el entrenamiento en la toma de decisiones en situaciones de alta presión y manejo de complicaciones intraoperatorias. La literatura sugiere que, aunque el TurpMentor brinda una excelente base técnica, el realismo mejorado podría hacer que el entrenamiento sea aún más efectivo. Esta revisión se enmarca en una metodología descriptiva con un enfoque cualitativo, que se enfoca en analizar los datos de un grupo de 20 cirujanos que han utilizado el TurpMentor para su entrenamiento. Se estudiaron los patrones de desempeño y su relación con la recuperación postoperatoria de los pacientes en un entorno clínico. La evaluación de la correlación entre las habilidades adquiridas en el simulador y los resultados clínicos mostró que, en general, un mayor rendimiento en el simulador se correlaciona con una mejor recuperación de los pacientes, en términos de menor tiempo de hospitalización y reducción de complicaciones postoperatorias. El uso del TurpMentor demuestra ser una herramienta educativa valiosa, especialmente en la fase inicial del aprendizaje

quirúrgico. Sin embargo, la falta de simulación realista de ciertos factores, como el sangrado, indica que el entrenamiento completo debe complementarse con otras experiencias prácticas para fortalecer la competencia del cirujano en entornos clínicos reales. Se concluye que estos simuladores pueden desempeñar un rol crucial en la educación médica moderna, siempre y cuando se reconozcan y compensen sus limitaciones.

En la investigación [25], menciona que el uso de simuladores de realidad virtual (RV), como el PercMentor y el MariónK81 en combinación con dispositivos como el Meta Quest 2, representa un avance en la capacitación médica al ofrecer gráficos detallados y alta fidelidad visual, elementos que potencian el realismo y la inmersión en el entrenamiento quirúrgico. La recopilación de datos en tiempo real permite analizar aspectos críticos del rendimiento de los usuarios, facilitando un seguimiento detallado del progreso en habilidades específicas. Sin embargo, estos simuladores aún enfrentan desafíos para replicar procedimientos abiertos, especialmente en lo que respecta a la simulación de tacto y resistencia de tejidos, debido a limitaciones en la retroalimentación háptica. Esta carencia impacta en la experiencia de aprendizaje al reducir la percepción de sensaciones físicas, esenciales en la cirugía real. La investigación en este ámbito sigue una metodología investigativa que combina análisis cuantitativo y cualitativo, permitiendo evaluar la efectividad de los simuladores y su impacto en la formación médica. Este enfoque busca validar los beneficios de estos sistemas y optimizar su aplicabilidad para una formación médica integral en procedimientos de alta complejidad.

De acuerdo con [26], los modelos físicos de simuladores de realidad virtual (RV), como los dispositivos Meta Quest 2, ofrecen un entorno inmersivo que permite a los profesionales de la salud practicar técnicas quirúrgicas en un contexto controlado. Los gráficos detallados y la alta fidelidad visual en realidad virtual (RV) son elementos clave para la efectividad de los simuladores médicos, **especialmente** en entornos de cirugía. Al utilizar texturas avanzadas, sombreado realista e iluminación dinámica, los desarrolladores de RV logran replicar con precisión la apariencia y textura de tejidos, órganos y herramientas quirúrgicas, lo cual incrementa el realismo y la inmersión del usuario. Esto no solo facilita que los profesionales de la salud se familiaricen con el entorno quirúrgico, sino que también mejora la retención de habilidades técnicas. En

el análisis de resultados muestra que estos simuladores enfrentan el desafío de ofrecer herramientas validadas para procedimientos abiertos, donde el realismo y la precisión son fundamentales. Sin embargo, la falta de retroalimentación háptica avanzada en muchos de estos simuladores de RV sigue siendo una limitación, ya que dificulta la experiencia táctil y la percepción de resistencia que ocurren en situaciones quirúrgicas reales. La metodología en el desarrollo de aplicaciones de RV volumétricas para simulación quirúrgica, como en el caso del proyecto VolumétricosOR, adopta un enfoque interdisciplinario que integra áreas como la cirugía, el diseño de simuladores y los estudios de medios, buscando una experiencia educativa integral y realista. En el estudio comparativo entre la RV para desarrollar habilidades básicas en cirugía mínimamente invasiva y el entrenamiento robótico en simulación quirúrgica con el simulador RobotTixMentor, se analizó el desempeño de 26 participantes sin experiencia previa en cirugía. Los participantes fueron asignados aleatoriamente a dos grupos: el grupo de procedimientos en RV (n = 13), enfocado en habilidades básicas y orientado a una experiencia inmersiva en procedimientos, y el grupo de simulación de RV básica (n = 13), que se concentró en habilidades fundamentales sin el contexto completo de un procedimiento quirúrgico.

Según el estudio desarrollado por [27] la realidad virtual inmersiva (IVR) y la realidad mixta (RM) representan un espectro dentro de la tecnología de realidad extendida (XR), que está revolucionando la formación quirúrgica moderna, especialmente en áreas como la ortopedia. Estas tecnologías combinan el entorno real con simulaciones virtuales avanzadas, permitiendo a los cirujanos interactuar con modelos anatómicos 3D en tiempo real, lo cual mejora la precisión y facilita el aprendizaje de técnicas complejas en un entorno controlado. La literatura revisada destaca que tanto la IVR como la RM permiten una práctica quirúrgica que no depende exclusivamente del paciente, reduciendo riesgos y mejorando la retención de habilidades a través de la repetición. Sin embargo, uno de los principales desafíos de estas tecnologías es evaluar la relación coste-eficacia, especialmente dado el costo de desarrollo y mantenimiento de simuladores de alta fidelidad. Hasta la fecha, la investigación se ha centrado en el rendimiento quirúrgico y en los beneficios educativos de estos sistemas, pero aún falta realizar estudios exhaustivos que analicen si estos beneficios justifican la inversión inicial y los costos de implementación a largo plazo. La metodología investigativa

empleada en la mayoría de los estudios sobre IVR y RM es mixta, incorporando métodos cuantitativos para medir el desempeño técnico (precisión, velocidad y error en movimientos) y métodos cualitativos para evaluar la satisfacción y percepción del usuario. Esta combinación metodológica es clave para proporcionar una visión completa del impacto de estas tecnologías en la formación quirúrgica y para guiar futuras decisiones sobre su adopción en programas de enseñanza médica.

La propuesta de [28], comenta en su artículo que el uso de realidad virtual inmersiva (IVR) para la enseñanza de habilidades quirúrgicas, las grabaciones del mundo real han demostrado ser una herramienta efectiva para capacitar a estudiantes en técnicas básicas de sutura, incluso en aquellos sin formación médica previa. Un estudio reciente evaluó la eficacia de este método con una población de 44 estudiantes preuniversitarios, asignados al azar para recibir entrenamiento en IVR. Utilizando una metodología descriptiva con enfoque cuantitativo, se aplicaron pruebas estadísticas ANOVA y T para analizar las diferencias en el rendimiento y la retención de habilidades entre los estudiantes. Los resultados revelaron mejoras significativas en la precisión y el tiempo de ejecución de suturas, lo que sugiere que la IVR puede facilitar el aprendizaje de competencias técnicas básicas. Para las puntuaciones de sutura, Los resultados del estudio mostraron que los estudiantes entrenados con realidad virtual inmersiva (IVR) lograron puntuaciones de sutura significativamente mayores que el grupo que utilizó el manual escrito, con una diferencia estadísticamente significativa ($p = 0,031$) y un tamaño del efecto considerable (D de Cohen = 0,92), lo cual indica una mejora notable en la habilidad técnica. Sin embargo, la ventaja de la IVR fue menos pronunciada al compararla con el grupo entrenado con videos en 2D, donde la diferencia no fue estadísticamente significativa ($p = 0,16$) y el tamaño del efecto fue menor (D de Cohen = 0,65). Estos resultados sugieren que, si bien la IVR ofrece ventajas importantes en la adquisición de habilidades quirúrgicas básicas en comparación con métodos tradicionales de aprendizaje textual, su beneficio adicional sobre el video 2D es menos marcado. Esta investigación aporta evidencia sobre la efectividad de las grabaciones de procedimientos reales en IVR como un complemento en la formación inicial, permitiendo la práctica segura y controlada de habilidades esenciales para futuros profesionales de la salud.

En su publicación [29] sugiere un enfoque investigativo cualitativo para evaluar la integración de tecnologías inmersivas, como la realidad virtual (RV), en la educación médica. A través de una revisión de literatura y encuestas aplicadas a estudiantes de medicina, el estudio se centra en analizar cómo estos entornos tridimensionales, dispositivos interactivos y métodos de video HDM 360 en 3D como Oculus y HTC VIVE en el aprendizaje de medicina pueden complementar el aprendizaje tradicional. Los resultados indican que el 91 % de los estudiantes considera fundamental integrar estas tecnologías en la enseñanza, mientras que el 70 % valora que estas herramientas fomentan un aprendizaje centrado en el estudiante. Además, el 77 % de los encuestados está de acuerdo en que la educación virtual mejora la participación en el aprendizaje. Este tipo de investigación resalta el valor de estas tecnologías como complemento pedagógico, sin desplazar las metodologías tradicionales, y destaca su papel en el aprendizaje activo y en la mejora de la experiencia educativa.

En su publicación [30], reseña que la tecnología de RV ha experimentado avances significativos en los últimos años, ofreciendo experiencias cada vez más inmersivas y detalladas, especialmente útiles en el ámbito de la capacitación médica. La RV inmersiva en 3D permite a los estudiantes de medicina y a los profesionales de la salud practicar habilidades, observar procedimientos y explorar la anatomía en un entorno controlado y seguro que simula situaciones clínicas reales. Este tipo de capacitación es particularmente valioso porque permite la repetición, el aprendizaje a través de errores sin riesgo para los pacientes y la adaptación a escenarios específicos que pueden ser difíciles de replicar en la vida real. También, reseña que la investigación cualitativa realizada con revisión literaria ha demostrado que la RV en educación médica se percibe como una herramienta de necesidad oportuna. Proporciona un enfoque innovador para la enseñanza, mejorando la retención de conocimientos y habilidades, y se está posicionando como un recurso que complementa de manera óptima la educación tradicional. La RV inmersiva en 3D en medicina no solo facilita una comprensión más profunda de conceptos complejos, sino que también apoya la formación de habilidades prácticas críticas, preparando mejor a los futuros profesionales de la salud.

En el trabajo de [31] se aborda la tecnología inmersiva, particularmente la realidad virtual (RV), ha mostrado un gran potencial para aumentar la eficacia del aprendizaje

quirúrgico. Estas herramientas permiten a los estudiantes y residentes practicar procedimientos complejos en un entorno controlado y seguro, mejorando su habilidad y precisión antes de enfrentarse a casos reales. El modelo de enseñanza Brief-Intraoperative Teaching-Debrief (BID) y el método One-Minute Preceptor, que son enfoques que pueden integrarse con tecnologías de RV para estructurar la enseñanza quirúrgica de manera eficiente y efectiva. BID se enfoca en guiar al estudiante antes y después de un procedimiento, permitiendo la reflexión crítica y la corrección de errores, mientras que One-Minute Preceptor ofrece retroalimentación rápida y específica, reforzando el aprendizaje en momentos claves de la práctica quirúrgica. Combinados con RV, estos modelos mejoran la retención del conocimiento, aumentan la seguridad en el quirófano y ayudan a los estudiantes a adquirir destrezas prácticas con mayor rapidez y confianza.

1.3 Fundamentación teórica

1.3.1 Realidad Virtual en la Educación Médica

La realidad virtual (RV), ha evolucionado como una herramienta potente en la educación médica, permitiendo a los estudiantes experimentar entornos de aprendizaje inmersivos y simulados sin los riesgos inherentes al contacto con pacientes reales. Según Xi Li y Dalia Elnagar, la RV ha mostrado ser especialmente beneficiosa en países de ingresos medios y bajos, donde el acceso a simulaciones clínicas puede estar limitado [1]. En entornos educativos, la RV ofrece una repetición de prácticas en escenarios clínicos y quirúrgicos, lo que mejora la retención de conocimientos y la precisión en el aprendizaje. Diversos estudios indican que la integración de la RV en el currículo medico fomenta una adquisición acelerada de habilidades prácticas y aumenta la confianza en los estudiantes [2]. La interacción en tiempo real en entornos virtuales también facilita el aprendizaje a través de la retroalimentación inmediata, esencial en la educación práctica.

1.3.2 Simulación de Emergencias y Entrenamiento en Habilidades Clínicas

La simulación de escenarios de emergencia mediante RV es fundamental para preparar a los estudiantes ante situaciones críticas. La RV permite recrear entornos de primeros auxilios y emergencias, ofreciendo un espacio seguro donde los estudiantes pueden desarrollar habilidades críticas y evaluar su rendimiento bajo presión. Según la investigación de Meyerheim y colaboradores, los entornos de emergencia en RV han mejorado la capacidad de respuesta en estudiantes de medicina al replicar factores estresantes y complejidades cínicas difíciles de lograr en simulaciones físicas tradicionales [3].

1.3.3 Ventajas de la Realidad Virtual frente a Métodos Tradicionales

En comparación con los métodos de enseñanza tradicionales, la RV presenta varias ventajas importantes. A diferencia de los laboratorios físicos, que pueden ser costosos y limitados en capacidad, la RV permite a los estudiantes practicar de manera ilimitada, sin restricciones de espacio ni de recursos físicos. Según Pedram y Ogie, los costos iniciales de la RV se ven compensados por la reducción de gastos a largo plazo en infraestructura física, así como por la mejora en la calidad educativa [4]. Otra ventaja es la accesibilidad y la capacidad de repetición, permitiendo que los estudiantes practiquen múltiples veces sin depender de la disponibilidad de recursos físicos o supervisión constante.

1.3.4 Desafíos y Limitaciones en el Uso de Realidad Virtual en Educación Médica

A pesar de los beneficios, la implementación de la RV en la educación médica enfrenta desafíos técnicos y financieros. Los costos iniciales de los equipos de RV, la capacitación requerida para su uso y las limitaciones en la fidelidad de algunos simuladores son obstáculos comunes. Mergen y Graf mencionan que la adaptabilidad del software y hardware sigue siendo una barrera, particularmente en simuladores que requieren una alta precisión en la retroalimentación táctil [3]. Además, algunos estudios resaltan la curva de aprendizaje para los estudiantes, que puede alargar el

tiempo necesario para obtener los beneficios plenos de la simulación. Sin embargo, los avances continuos en la tecnología RV y la reducción de costos auguran una integración más sencilla en el futuro.

1.3.5 La realidad virtual y la realidad aumentada

Para definir el término de RV, como aquella simulación de un entorno real que puede experimentar visualmente en un ámbito espacial proveyendo una interacción de movimiento posiblemente incluya dispositivos hápticos, cascos guantes para la interacción [32]. Por su parte, para [33] la RV, es una simulación de un ambiente tridimensional generada por computadoras, en el que el usuario es capaz tanto de ver como de manipular los contenidos de ese ambiente. La RV puede ser utilizada en áreas como: entretenimiento, educativos e industriales Existen dos variantes de la RV:

- RV inmersiva: no utiliza ningún dispositivo que sumerja al usuario dentro del mundo virtual.
- RV no inmersiva. Utiliza dispositivos del tipo Head Mounted Display (HMD).

Por otra parte, existe el término de RA, que ofrece una percepción e interacción con el mundo real, brindando al usuario un escenario real, incrementándose con datos adicionales generados por computación. De este modo, la realidad física se combina con elementos virtuales, disponiéndose de una realidad mixta en tiempo real [34]. La RV mixta, se reconoce como realidad RA cuando incorpora objetos digitales virtuales en una escena real tridimensional, o como RV que incluye elementos del mundo real en un entorno virtual [35].

1.3.6 Funcionamiento del mundo virtual

Las interacciones de los usuarios son registradas en tiempo real por los dispositivos o sensores. De acuerdo con el sistema de visualización se requiere posicionar la cabeza del usuario, aunque también es frecuente capturar movimientos de la mano y de los dedos. Esta información es enviada al computador, donde es procesada por los módulos de recogida y tratamiento de datos de entrada. Cuando las interacciones de los usuarios están en un formato adecuado, se utiliza la simulación, tanto para

introducir posibles cambios en el mundo virtual como para generar imágenes, sonidos y otros datos para completar la simulación sensorial [35]. En el caso de la RA se integran el mundo real (físico) y el mundo virtual. Para que la RA pueda ejecutarse, se necesitan tres parámetros esenciales:

- Referencia un objeto real para la interpretación y creación del objeto virtual.
- La presencia de un dispositivo con cámara, cm un smartphone para transmitir la imagen del objeto real.
- Un algoritmo por capturar la señal transmitida por la cámara.

Por medio de la cámara, el objeto real capturado por el algoritmo, que recibe la imagen y la combina con proyecciones 3D. Una vez proyectadas son agregadas en la imagen y sobrepuestas en el ámbito físico, reflejando el resultado de la RA al usuario.

1.3.7 La realidad virtual en el marco de la enseñanza médica

Estas tecnologías son de uso reciente en el área educativa. aunque se ha comprobado sus efectos positivos en el aprendizaje, todavía, se encuentra en una etapa inicial [36]. En el sector del aprendizaje médico la RV y la RA se encuentra cada vez más arraizada. Por medio de la simulación con la RV, la RA y la RM, los estudiantes en el área médica aprenden en mundos virtuales y practican situaciones de la vida real sin riesgo alguno. Estos tienen el potencial de proporcionar experiencias como reales, es así como, un grupo puede visitar una ciudad en otro país, desde el salón de clase haciendo la experiencia accesible y financieramente viable [35]. Permitiendo señalar que la interacción con enseñanza no resulta difícil ni compleja, que pueden ser útiles para aumentar su aprendizaje y que les permitiría la comprensión de ciertos conceptos [37].

1.3.8 Motores para desarrollo de VR

Los motores de desarrollo son plataformas conformadas por un conjunto de herramientas que permiten crear, desarrollar, diseñar y ejecutar aplicaciones inmersivas en entornos virtuales. Estos motores facilitan la creación de objetos gráficos en 3D, la realización de simulaciones físicas, interacciones, entre otros. Son

utilizados en diferentes áreas, desde los videojuegos hasta la enseñanza educativa. A continuación, se presentan los motores de desarrollo más populares para crear realidad virtual en la actualidad.

- **Unity:** Es un motor gráfico en 3D utilizado para la creación de juegos en 2D y 3D, aplicaciones interactivas, animaciones tridimensionales y visualizaciones. Aunque puede emplearse para modelado, su enfoque principal está en el desarrollo de interacciones inmersivas, por lo que su uso en modelado es limitado [38]. Para desarrollar en Unity, se trabaja con escenarios y se utiliza el lenguaje de programación C# mediante el framework NET. Dicho lenguaje guarda similitudes con Java, lo que permite aplicar programación orientada a objetos. Unity es uno de los motores más populares entre la comunidad de desarrolladores de videojuegos, lo que garantiza un amplio soporte y recursos para resolver problemas durante el desarrollo del sistema [39].

- **Unreal Engine:** Es un motor de desarrollo creado por la empresa Epic Games. Se trata de un software multiplataforma ampliamente utilizado por desarrolladores de videojuegos, que compite directamente con Unity. Está programado en C++ y, como cualquier motor, se encuentra en constante evolución para adaptarse a los avances tecnológicos actuales, incluyendo soporte para realidad virtual. Su capacidad de trabajar en tiempo real, permitiendo a los desarrolladores visualizar simultáneamente el diseño, la ejecución y las interacciones implementadas [40]. En cuanto a su modelo de negocio, Unreal Engine ofrece una licencia gratuita con ciertas restricciones. Sin embargo, incluso en su versión sin costo permite crear sistemas de alta calidad. Cuando un producto supera los 3,000 USD de ganancias trimestrales, el desarrollador debe pagar a Epic Games un 5% de regalías sobre los ingresos generados [40].

- **Godot:** Es un motor de desarrollo para la creación de videojuegos en 2D y 3D, caracterizado por ser multiplataforma y de código abierto. Soporta múltiples lenguajes de programación, incluyendo GDScript (su lenguaje nativo), Visual Scripting, C# y C++, lo que ofrece flexibilidad a los desarrolladores. Al ser completamente libre, permite publicar los proyectos en cualquier plataforma

sin restricciones de licencias. Una de sus características distintivas es el sistema de Viewports, funcionalidad que no está presente en otros motores. Los Viewports son áreas rectangulares donde se renderiza el mundo del juego. Su constante desarrollo de nuevas funcionalidades lo posicionan como una alternativa sólida frente a motores comerciales [41].

- **CryEngine:** Es un motor de desarrollo de videojuegos creado por la empresa Crytek. Se caracteriza por ser una herramienta de alta calidad gráfica en tiempo real, lo que exige un mayor uso de recursos hardware para generar experiencias inmersivas con un alto nivel de detalle. Su rendimiento visual lo hace ideal para proyectos que requieren realismo avanzado, como entornos abiertos y efectos de iluminación dinámica. El lenguaje de programación utilizado en CryEngine es C++, el cual, al pertenecer a la familia de lenguajes C, permite el uso de POO, esto facilita el desarrollo de sistemas complejos y optimizados, aprovechando al máximo el rendimiento del motor [38]. En la actualidad, CryEngine es multiplataforma, lo que representa una ventaja significativa para los desarrolladores. A diferencia de sus primeras versiones, que solo eran compatibles con Microsoft Windows, PlayStation 3 y Xbox 360, ahora soporta una mayor variedad de dispositivos. Uno de los avances más destacados de CryEngine fue su implementación en la actualización del juego Hunt: Showdown para PlayStation 4, así como en el desarrollo de la reconocida saga Far Cry [40].

1.3.9 Lenguajes de Programación

De acuerdo con [42], los lenguajes de programación más utilizados para realidad virtual son C#, C++, Python, JavaScript, Java y Swift:

- **C#:** Es uno de los lenguajes más empleados en el desarrollo de aplicaciones de realidad virtual, especialmente en Unity. Utiliza el framework .NET y combina características de C++ y Java. Como lenguaje de código abierto, destaca por su seguridad y facilidad de uso. Se aplica en el desarrollo de aplicaciones de escritorio, móviles y videojuegos [42].

- **C++:** Este lenguaje es fundamental para trabajar con Unreal Engine, uno de los motores más potentes del mercado. Aunque presenta una curva de aprendizaje alta, su eficiencia, escalabilidad y capacidad de reutilizar bloques de código lo hacen ideal para proyectos complejos. Pertenece a la familia de lenguajes C y se considera un lenguaje de alto nivel [42].

- **Java:** Como lenguaje orientado a objetos, Java permite desarrollar aplicaciones de realidad virtual compatibles con múltiples dispositivos como Leap Motion, HTC Vive/Oculus y Google Cardboard. Su portabilidad se debe al uso de la Java Virtual Machine (JVM), que ejecuta el código compilado en cualquier plataforma [42].

- **Python:** Reconocido como uno de los lenguajes más populares [43], Python destaca por su versatilidad. Además de aplicaciones de realidad virtual, se emplea en análisis de datos y desarrollo web. Al ser interpretado, no requiere compilación, y su sintaxis clara (que usa tabulaciones en lugar de llaves) lo hace accesible para principiantes [44].

- **JavaScript:** Esencial para la programación web, JavaScript permite crear experiencias de realidad virtual mediante plataformas como WebVR. Su simplicidad y compatibilidad con bibliotecas como Three.js facilitan el desarrollo de entornos 3D interactivos en navegadores [44].

- **Swift:** Desarrollado por Apple, este lenguaje multiplataforma, fuertemente tipado y de código abierto se utiliza para crear aplicaciones nativas en sus sistemas operativos. También soporta el desarrollo de realidad virtual y aumentada, permitiendo la integración de objetos 3D en entornos gráficos. Como lenguaje de alto nivel, combina eficiencia con una sintaxis legible [44].

1.3.10 Herramientas de Modelado 3D

El modelado tridimensional representa una fase fundamental en la creación de entornos virtuales. Estas aplicaciones permiten diseñar los objetos, personajes y escenarios que posteriormente se integrarán en los proyectos de RV. Sin ellas,

resultaría imposible alcanzar los niveles de realismo que demandan las experiencias inmersivas actuales. Las herramientas de modelado más usadas son:

- **Blender:** Es un software open-source que se ha convertido en una opción profesional accesible para creadores 3D. Ofrece herramientas que van desde modelado básico hasta procesos avanzados como rigging, animación y renderizado fotorrealista. Gracias a su gran comunidad de usuarios, se utiliza ampliamente en la industria del entretenimiento y el diseño digital [45].

- **Autodesk Maya:** Es considerado el estándar industrial en muchas producciones AAA (videojuegos de alto presupuesto y gran repercusión, producidos por grandes editoriales.) ofrece un conjunto de funciones profesionales para modelado, animación compleja y simulaciones dinámicas [46].

- **Cinema 4D:** Es una solución de software profesional de modelado, animación, simulación y renderizado en 3D. Rápido, potente, flexible y estable, este potente conjunto de herramientas hace que los flujos de trabajo en 3D sean más accesibles y eficientes para diseño, gráficos en movimiento, efectos visuales, realidad aumentada, realidad mixta y realidad virtual, desarrollo de juegos y todo tipo de visualización [47].

1.3.11 Complejidad en el Desarrollo de Entornos de Realidad Virtual para Educación en Enfermería

El desarrollo de aplicaciones en realidad virtual (RV) representa un alto nivel de complejidad técnica, especialmente porque aún no existen metodologías consolidadas y específicas que guíen de forma estructurada todo el proceso de creación. A diferencia del desarrollo de software tradicional, donde existen marcos metodológicos establecidos como Scrum, XP o Kanban, en el ámbito de la RV no se dispone de modelos de desarrollo adaptados a las particularidades de los entornos inmersivos. Esto significa que el diseño, implementación y evaluación de estas experiencias depende en gran medida del criterio del equipo de desarrollo, lo que introduce un margen considerable de subjetividad y riesgo técnico. Esta ausencia de estándares

metodológicos afecta todas las etapas del proyecto, desde la planificación de la experiencia inmersiva hasta la integración del contenido educativo y la validación de la interacción del usuario. En el presente proyecto, se busca simular el armado de la mesa mayo dentro de un entorno quirúrgico, esta complejidad se incrementa por la falta de recursos especializados. Debido a la ausencia de un modelador 3D, no fue posible representar tridimensionalmente el instrumental quirúrgico, por lo que se optó por utilizar fotografías reales de elementos como tijeras, pinzas y separadores, adaptadas a un entorno interactivo en RV. Esta solución, aunque funcional, también implica una adaptación creativa del flujo de desarrollo, evidenciando la necesidad urgente de metodologías que guíen el diseño instruccional, la fidelidad visual, la ergonomía virtual y la usabilidad [48], [49].

Desde la perspectiva técnica, se presentan desafíos importantes tanto en el ámbito del software como del hardware. A nivel de software, se requiere dominio de motores de desarrollo de realidad virtual como Unity, implementación de sistemas de interacción mediante herramientas como XR Interaction Toolkit o OVRPlugin, seguimiento de manos y desarrollo de interfaces adaptadas al entorno inmersivo. La incorporación de imágenes en lugar de modelos tridimensionales obligó a diseñar un sistema visual que permitiera al usuario identificar, clasificar y manipular fotografías bidimensionales en un espacio tridimensional, lo cual requiere una lógica adicional en cuanto a colisiones, interacciones y retroalimentación visual. Por otro lado, el hardware no solo incluye los visores como Meta Quest 3, sino también computadoras con características muy específicas: procesadores de alto rendimiento (por ejemplo, Intel Core i7 o AMD Ryzen 7), tarjetas gráficas dedicadas (como NVIDIA GTX 1050Ti o superior), al menos 16 GB de memoria RAM y almacenamiento SSD para manejar eficientemente los tiempos de carga y renderizado. Estos requisitos son indispensables para garantizar una experiencia de desarrollo fluida y una ejecución sin interrupciones, ya que cualquier deficiencia en el rendimiento puede provocar errores gráficos, caídas de fotogramas o incluso la imposibilidad de compilar entornos complejos de RV [50], [51]. La creación de experiencias educativas en RV, como la que se propone para estudiantes de enfermería, exige un equilibrio entre realismo visual, funcionalidad precisa y coherencia pedagógica. Aunque no se contaba con modelos 3D del instrumental, se logró una solución efectiva utilizando fotografías categorizadas por

tiempos quirúrgicos (corte, hemostasia, tracción, disección/separación y sutura). Esta aproximación permitió al usuario interactuar con los instrumentos de manera lógica y funcional, reforzando el conocimiento a través de una experiencia inmersiva. No obstante, este tipo de adaptación demuestra que el desarrollo de RV en educación médica requiere soluciones creativas ante las limitaciones técnicas y metodológicas, lo que subraya la necesidad de enfoques multidisciplinarios y de nuevas guías de desarrollo orientadas a este campo [49], [52].

1.3.12 Métricas para medir complejidad de desarrollo

Las métricas de complejidad de software son usadas por los desarrolladores para analizar cuán difícil es leer el código fuente de un programa. Además, pueden evaluar la facilidad de mantenimiento y cuánto dependen los módulos de sistema entre sí. Algunas métricas incluyen la Cyclomatic Complexity, Halstead Complexity, entre otras. Las métricas son cruciales para identificar áreas del código que son propensas a errores, son difíciles de escalar o que necesitan refactorización, lo que, al hacerlo, eleva la calidad y la eficiencia de la programación.

1.3.13 Métricas de Rendimiento en Tiempo Real

Las métricas de rendimiento en tiempo real se utilizan para medir cómo se comporta un software mientras está en ejecución, especialmente en sistemas interactivos como videojuegos, simulaciones o aplicaciones críticas. Estas métricas incluyen FPS (Frames Per Second) y Uso de CPU/GPU y nos permiten monitorear aspectos como el uso del procesador (CPU), la carga de la tarjeta gráfica (GPU), el consumo de memoria y el tiempo de respuesta. Gracias a estas mediciones, los desarrolladores pueden identificar cuellos de botella, optimizar procesos y garantizar una experiencia fluida y estable para el usuario final, incluso bajo condiciones de alta demanda.

1.3.14 Métricas de Experiencia de Usuario (UX)

Las métricas de Experiencia de Usuario (UX) constituyen un componente fundamental en la evaluación de sistemas interactivos, ya que permiten analizar cómo los usuarios perciben y se relacionan con una interfaz o entorno digital. Entre estas métricas se

encuentra el tiempo de respuesta a los inputs, el cual mide la rapidez con la que el sistema responde a las acciones del usuario, siendo un indicador clave de eficiencia y fluidez en la interacción. Otra herramienta ampliamente utilizada es la encuesta SUS (System Usability Scale), que consiste en un cuestionario estandarizado para evaluar la usabilidad percibida, proporcionando resultados cuantificables a partir de la opinión del usuario. Por otra parte, para medir la carga de trabajo experimentada durante el uso del sistema, se utilizan instrumentos como el método NASA-TLX, que evalúa dimensiones como la demanda mental, física y temporal, el esfuerzo, el rendimiento percibido y el nivel de frustración. Estas métricas, en conjunto, proporcionan una visión integral sobre la calidad de la experiencia interactiva, permitiendo identificar oportunidades de mejora en sistemas orientados a la interacción humano-computadora.

1.3.15 Metodologías de desarrollo

Las metodologías de desarrollo de software son marcos estructurados que establecen prácticas, técnicas y procedimientos para guiar el proceso de creación de sistemas informáticos desde su concepción hasta su implementación y mantenimiento. Existen diversas metodologías de desarrollo, que ofrecen enfoques estructurados según las características del proyecto, el entorno y las necesidades del cliente. Estas metodologías se pueden agrupar principalmente en dos grandes categorías: tradicionales y ágiles.

1.3.16 Metodologías Tradicionales

Las metodologías tradicionales, también conocidas como predictivas, siguen un enfoque secuencial y estructurado. Estas metodologías asumen que los requisitos del proyecto pueden ser definidos completamente al inicio y que los cambios durante el desarrollo deben minimizarse. Son adecuadas para proyectos donde los requisitos son estables y bien entendidos.

Algunas metodologías tradicionales relevantes son:

- **Modelo en Cascada (Waterfall):**

  Enfoque secuencial donde cada fase debe completarse antes de pasar a la siguiente: requisitos, diseño, implementación, verificación y mantenimiento. Ideal para proyectos con requisitos claros y bien definidos desde el principio [53].

- **Rational Unified Process (RUP):**

  Marco de desarrollo iterativo que divide el proyecto en cuatro fases (inicio, elaboración, construcción, transición) con disciplinas específicas para cada fase. Usa UML como lenguaje de modelado [54].

- **Modelo en V:**

  Extensión del modelo en cascada donde cada fase de desarrollo tiene una fase correspondiente de verificación. Enfatiza la validación temprana y las pruebas [55].

- **Modelo en Espiral:**

  Combina elementos de desarrollo iterativo con aspectos sistemáticos del modelo en cascada. Cada iteración pasa por cuatro fases: planificación, análisis de riesgos, ingeniería y evaluación [56].

1.3.17 Metodologías Ágiles

Las metodologías ágiles surgieron como respuesta a la rigidez de los modelos tradicionales, especialmente en entornos donde los requisitos cambian frecuentemente. Su enfoque está en la flexibilidad, colaboración continua con el cliente y entregas incrementales de valor funcional [57].

Algunas metodologías ágiles destacadas son:

- **Scrum:**

  Scrum es un marco de trabajo ágil que permite a los equipos trabajar de forma colaborativa en ciclos cortos llamados "sprints". Se caracteriza por roles específicos (Scrum Máster, Product Owner, Equipo de Desarrollo), eventos (Sprint Planning, Daily Scrum, Sprint Review) y artefactos (Product Backlog, Sprint Backlog) [58].

- **Extreme Programming (XP):**

  XP enfatiza la calidad del código y la respuesta rápida a los cambios de requisitos. Sus prácticas incluyen programación en parejas, desarrollo guiado por pruebas (TDD), integración continua y refactorización constante [59].

- **Kanban:**

  Originado en el sistema de producción Toyota, Kanban visualiza el flujo de trabajo, limita el trabajo en progreso y optimiza el flujo continuo de entrega. Se centra en entregar valor al cliente de manera constante [60].

- **Lean Software Development:**

  Adaptación de los principios Lean de manufactura al desarrollo de software. Enfatiza la eliminación de desperdicios, la entrega rápida, el empoderamiento del equipo y la optimización del conjunto [61].

1.4 Objetivos

1.4.1 Objetivo general

Implantar eficientemente una metodología para medir la complejidad del software mediante el uso de herramientas de realidad virtual utilizando instrumentos de evaluación.

1.4.2 Objetivos específicos

- Analizar el impacto de las herramientas de realidad virtual en la complejidad de los procesos de instrumentación ante una cirugía abierta por la capacidad limitada del laboratorio.
- Medir la complejidad del software desarrollado para simulaciones de realidad virtual mediante instrumentos de evaluación específicos, con el fin de identificar áreas de optimización y mejorar su eficiencia y funcionalidad.
- Desarrollar un sistema de realidad virtual para la usabilidad de instrumentación ante una cirugía abierta y así obtener un eficiente aprendizaje en la formación profesional de los estudiantes de enfermería dentro del laboratorio.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Los materiales utilizados en el desarrollo de este proyecto de investigación incluyeron diversos instrumentos diseñados para recopilar información de manera estructurada y confiable:

- Para conocer la percepción de los usuarios sobre la efectividad del sistema propuesto, se aplicaron encuestas específicas que permitieron recolectar datos cuantitativos (ver Anexo 1: Encuesta).

- Asimismo, se llevaron a cabo entrevistas dirigidas a docentes del área de la salud y experto en programación de realidad virtual, con el objetivo de obtener información sobre el funcionamiento del sistema, su aplicabilidad (ver Anexo 2: Entrevistas).

- Para la evaluación de la usabilidad del sistema, se utilizó una encuesta estandarizada de usabilidad, que permitió medir aspectos como la facilidad de uso, eficiencia, satisfacción del usuario y capacidad de aprendizaje del entorno desarrollado (ver Anexo 3: Encuesta de Usabilidad).

- En cuanto a la carga de trabajo cognitiva experimentada por los usuarios durante el uso del sistema, se empleó el método NASA-TLX, una herramienta validada ampliamente utilizada en estudios de ergonomía y usabilidad, que evalúa factores como la demanda mental, física y temporal, así como el esfuerzo, el rendimiento percibido y el nivel de frustración (ver Anexo 4: Cuestionario del método NASA TLX (Task Load Index) .

- Con el fin de medir la aceptación de la metodología, se aplicó un cuestionario basado en el modelo TAM (Technology Acceptance Model), el cual permitió analizar variables como la utilidad percibida y la facilidad de uso percibida, fundamentales para valorar la predisposición de los usuarios a adoptar la tecnología implementada (ver Anexo 5: Metodo de evaluación TAM).

2.2 Métodos

Para llevar a cabo esta investigación, se empleará un enfoque metodológico basado en el método cuantitativo, el cual permitirá medir y analizar de manera objetiva la relación entre las herramientas de realidad virtual y la enseñanza de prácticas de enfermería.

El diseño de la investigación es de tipo experimental y descriptivo, ya que no se manipularán variables directamente, sino que se observará y analizará el impacto de la implementación del sistema de realidad virtual en un entorno educativo real. Se recopilarán datos a través de encuestas y entrevistas aplicadas a los estudiantes de enfermería de quinto y sexto semestre de la Universidad Técnica de Ambato, quienes conforman la población de estudio.

Además, se utilizará un diseño transversal, dado que la recolección de datos se realizará en un solo punto en el tiempo, permitiendo evaluar la percepción y efectividad de la herramienta en el momento de su aplicación.

El método empleado facilitará la obtención de datos cuantificables para evaluar la usabilidad y funcionalidad del software, además de identificar posibles áreas de mejora en su implementación. La validación del sistema de realidad virtual se realizará a través de la comparación de métricas obtenidas en los cuestionarios aplicados a la muestra seleccionada, lo que permitirá fundamentar la toma de decisiones para optimizar el desarrollo del software.

2.2.1 Modalidad de la investigación

La investigación se fundamenta en un enfoque cuantitativo debido a la necesidad de medir, analizar y evaluar de manera objetiva la relación entre las variables. La variable independiente, herramientas de realidad virtual, influye directamente en la variable dependiente, complejidad del software, permitiendo establecer métricas precisas para evaluar cada etapa del desarrollo del sistema. Este enfoque garantiza la recolección de datos cuantificables, lo que resulta esencial para identificar las áreas de mejora en la estructura y funcionalidad del sistema, y así ayudar a resolver la falta de acceso continuo a prácticas esenciales en el laboratorio.

La investigación cuantitativa permite una evaluación rigurosa de las herramientas de realidad virtual, determinando cuál de ellas resulta más adecuada para optimizar el desarrollo del sistema. Este análisis incluye aspectos como la facilidad de implementación, escalabilidad y tiempo de desarrollo. Estas métricas proporcionan una base sólida para seleccionar herramientas que cumplan con los requisitos necesarios del sistema que contribuyan a una solución al problema.

La complejidad del software será medida mediante una metodología diseñada para analizar factores clave del proceso de desarrollo. Este enfoque cuantitativo permite medir aspectos como la estructura del código, los recursos utilizados, y los desafíos técnicos enfrentados en cada etapa. La recolección de datos numéricos facilita la comparación y el análisis de tendencias, asegurando que las decisiones de diseño e implementación estén fundamentadas en evidencia objetiva y orientadas a la mejora continua del sistema.

La implementación de una metodología asegura que los resultados sean replicables, verificables y aplicables en futuros proyectos de realidad virtual. Esto fortalece la validez del estudio, también contribuye al avance del conocimiento en el ámbito de la realidad virtual aplicada a la enseñanza. Al enfocar la investigación en datos objetivos y medibles, se garantiza que las soluciones propuestas sean efectivas, ayudando a las necesidades específicas de los estudiantes de la Facultad de Ciencias de la Salud.

2.2.2 Población y muestra

La población va a hacer los docentes y estudiantes de enfermería de quinto y sexto semestre de la Universidad Técnica de Ambato.

Quinto semestre:

- Paralelo A: 24 estudiantes
- Paralelo B: 30 estudiantes
- Paralelo C: 27 estudiantes

Sexto semestre:

- Paralelo A: 36 estudiantes
- Paralelo B: 33 estudiantes
- Paralelo C: 30 estudiantes

Docentes:

- 8 docentes

Total, de la probación: 188 personas.

Muestra:

n = tamaño de la muestra.
$N = tamaño\ de\ la\ población\ (180)$.
$Z = nivel\ de\ confianza\ (2.32)$.
$d = desviación\ estándar\ (0.5)$.
$e = error\ de\ estimación (0.08)$.

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

$$n = \frac{185 * 2.32^2 * 0.5 * (1 - 0.5)}{0.08^2 * (185 - 1) + 2.32^2 * 0.5 * (1 - 0.5)}$$

$$n = 99$$

De los 188 de nuestra población con un nivel de confianza del 98% y un error de estimación del 8% se obtiene una muestra de 99 personas.

2.2.3 Recolección de información

Los métodos de recolección de información son los siguientes:

Encuesta

El instrumento de recolección de datos consiste en una encuesta estructurada, diseñada para obtener información cuantitativa sobre "SISTEMA DE REALIDAD VIRTUAL DEDICADO A LA IDENTIFICACIÓN Y ARMADO DEL INSTRUMENTAL QUIRÚRGICO PARA ESTUDIANTES DE ENFERMERÍA", lo cual facilitó la recolección de datos de manera precisa, permitiendo a los participantes expresar sus opiniones con un mayor grado de detalle. Asegurando una medición confiable de las variables de interés. Además, el uso de esta escala favorece el análisis estadístico posterior, al proporcionar una base estructurada para interpretar los resultados. En total fueron 16 preguntas que se realizaron a los participantes, excluyendo la primera y la última dado que están son preguntas informales (ver Anexo 1: Encuesta).

Entrevistas

Se utilizaron entrevistas debido a que este método de investigación permite a los investigadores conocer las experiencias subjetivas de las personas. Estas facilitan la recopilación de datos detallados en forma de palabras, ideas y temas, lo que contribuye a obtener una comprensión más profunda del problema a resolver (ver Anexo 2: Entrevistas).

2.2.4 Procesamiento y análisis de datos

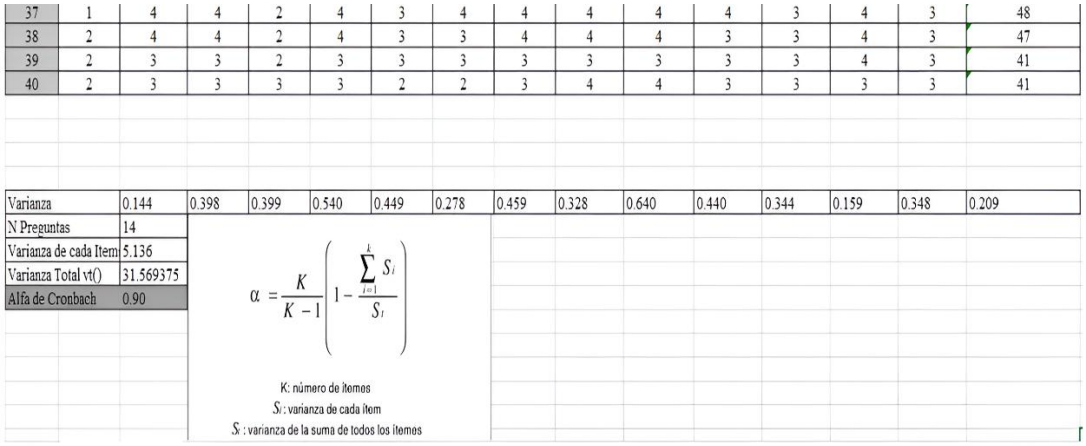
El procesamiento de datos se ejecutará con encuestas y entrevistas. Este paso permitirá estructurar los datos de manera coherente, facilitando su análisis posterior y garantizando la accesibilidad a la información relevante. Posteriormente, se procederá con la codificación de los datos, tanto cualitativos como cuantitativos, utilizando herramientas como Excel, SPSS u otras similares, permitiendo una sistematización efectiva que respalde el análisis estadístico y facilite la comparación de resultados entre distintas fuentes.

Para los datos cuantitativos (encuesta), se emplearán técnicas de estadística, como frecuencias, promedios y porcentajes, con el objetivo de identificar tendencias y patrones significativos. Para el análisis cualitativo (entrevistas), las respuestas abiertas se examinarán mediante análisis de contenido usando la herramienta ATLAS. Ti. El

objetivo es identificar temas comunes, patrones y opiniones clave que reflejen las percepciones de los participantes sobre el sistema de realidad virtual, con ello se obtendrán los requerimientos funcionales y no funcionales que va a tener el sistema. Los resultados se representarán a través de gráficos y tablas que faciliten su interpretación.

a. Validación de la Encuesta

Previo a la aplicación definitiva de la encuesta, se efectuó una validación del instrumento mediante un estudio piloto realizado con 40 estudiantes de quinto y sexto semestre de la carrera de Enfermería, seleccionados aleatoriamente. El análisis de consistencia interna, calculado con el coeficiente Alfa de Cronbach, registró un valor de 0.90 (ver Figura 1), de acuerdo con los parámetros de [62], este resultado refleja un nivel de confiabilidad excelente ($\alpha > 0.9$).



37	1	4	4	2	4	3	4	4	4	4	4	3	4	3	48
38	2	4	4	2	4	3	3	4	4	4	3	3	4	3	47
39	2	3	3	2	3	3	3	3	3	3	3	3	4	3	41
40	2	3	3	3	3	2	2	3	4	4	3	3	3	3	41

Varianza	0.144	0.398	0.399	0.540	0.449	0.278	0.459	0.328	0.640	0.440	0.344	0.159	0.348	0.209
N Preguntas	14													
Varianza de cada Item	5.136													
Varianza Total vt()	31.569375													
Alfa de Cronbach	0.90													

$$\alpha = \frac{K}{K-1}\left(1 - \frac{\sum_{i=1}^{k} S_i}{S_t}\right)$$

K: número de ítemes
S_i: varianza de cada ítem
S_t : varianza de la suma de todos los ítemes

Figura 1:Cálculo del Alfa de Cronbach de resultados obtenidos

b. Aplicación de la Encuesta

Comprobada la validez y confiabilidad del instrumento, la encuesta se administró a la población objetivo-compuesta por 188 personas. Se obtuvieron 174 respuestas válidas, ya que algunos estuvieron ausentes por motivos personales.

c. Análisis de resultados (Encuesta)

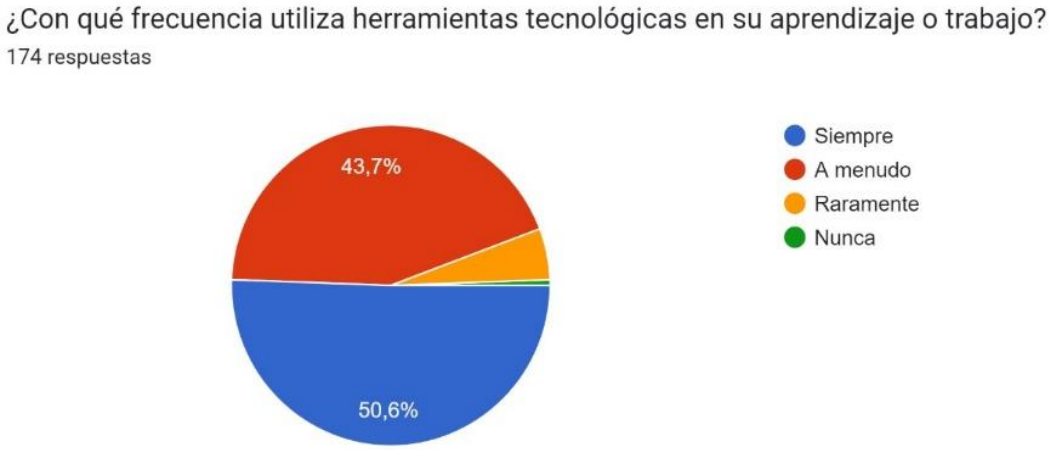


Figura 2: Frecuencia de uso de herramientas tecnológicas

En la Figura 2, la utilización de la tecnología por parte de los estudiantes de enfermería está representada en esta pregunta introductoria donde refleja que el 50, 6% integra las tecnologías en sus estudios y/o trabajo, seguidamente a menudo con el 43,7%, reflejando que una gran mayoría depende casi exclusivamente de las aplicaciones, herramienta y dispositivos tecnológicos para realizar sus actividades cotidianas, esto conlleva a estudiar aún los proyectos tecnológicos que tienen alguna relación con el aprendizaje en el contexto actual.

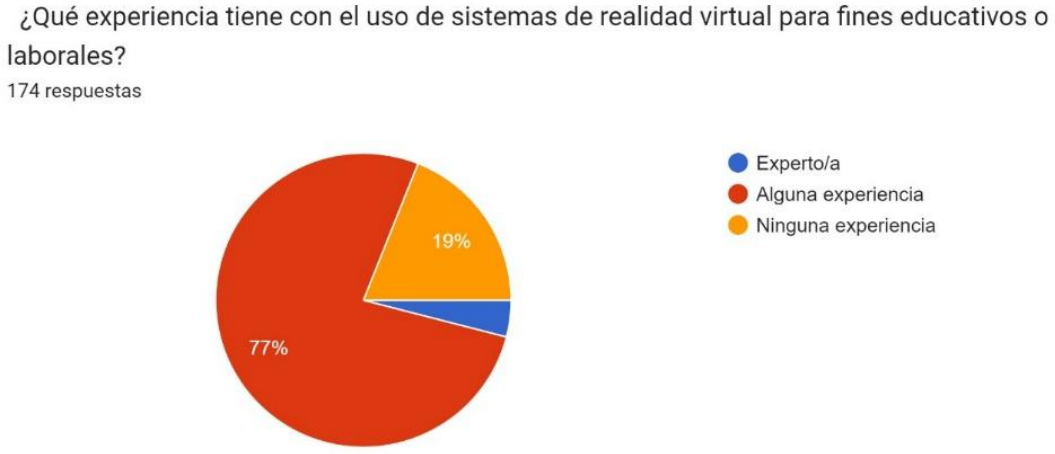


Figura 3: Uso de Realidad Virtual para Fines Educativos

En la Figura 3, como la mayoría de los encuestados respondieron en un 77%, son expertos o tienen alguna experiencia en el uso de sistemas virtualizados como la realidad virtual y aumentada. Esto hace sugerir que muchos estudiantes y/o

trabajadores tienen grandes conocimientos de estas plataformas tecnológicas y que en la actualidad está siendo muy empleadas por ellos en gran medida ante la posibilidad de simular entornos educativos con el fin de obtener mayores conocimientos o mejoras laborales. También reseña que solo el 4 % restante no ha tenido la oportunidad o experiencia en estas tecnologías, motivo por el cual debería realizarse un estudio separado con la finalidad de obtener información del porqué no la ha utilizado.

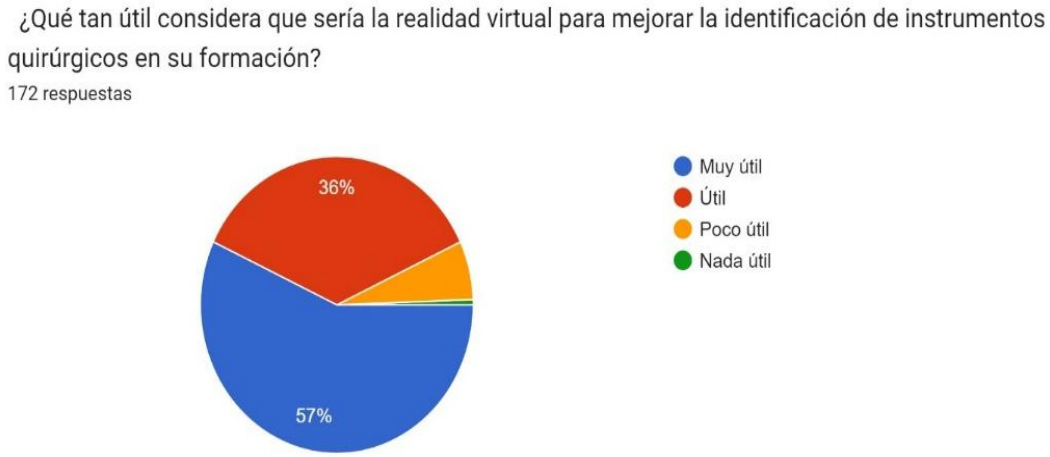


Figura 4: Percepción de la Utilidad de la Realidad Virtual en la Identificación de Instrumentos Quirúrgicos

En la Figura 4, los encuestados respondieron con un 57% muy útil, seguido de un 36% que es útil para la identificación de los instrumentos quirúrgicos sumando un 93% calificando a estas plataformas como una herramienta con un gran potencial en el aprendizaje médico-quirúrgico. Pero una minoría representada con el 7% considera que no son útiles o nada útil, esto se interpretaría que algunos estudiantes son suficientes con los procedimientos tradicionales o tienen dudas sobre su utilidad en situaciones reales, posibilitando muchos desafíos para su adopción en la educación médica.



Figura 5: Nivel de Satisfacción con las Herramientas Tecnológicas

En la Figura 5, más del 50% han considerado que el 179 de simuladores de sistemas virtualizados puede dar valor agregado a la enseñanza médica en un laboratorio físico junto al 36, 8%, no con el fin de sustituirlo sino de complemento tecnológico dentro del contexto del aprendizaje moderno. También hay que resaltar que 11,5% mantiene una desconfianza sobre el uso de estas tecnologías, sugiriendo una interpretación de no tener una mejor capacidad en un ambiente real.



Figura 6: Frecuencia de Uso: Aprendizaje en Línea vs. Presencial

En la Figura 6, el armado en el orden correcto de la mesa de mayo ha sido considerado por la gran mayoría de los encuestados como muy fácil o fácil de manejar mediante el uso de los sistemas virtualizados. Esto sugiere una valoración muy positiva para estas plataformas tecnológicas en su uso para la enseñanza médica. Aunque un poco más de la cuarta parte de los encuestados la consideran difícil o muy difícil, que pudiera

interpretarse como la falta de información de la utilidad de estas tecnologías, accesibilidad u optimización de sistemas implementados que no garantizan su eficiencia educativa, debido a la formación profesional de manera no tradicional.



Figura 7: Herramientas Tecnológicas Más Utilizadas en el Aprendizaje

En la Figura 7, el uso de la realidad virtual aumenta la confianza para identificar los instrumentos quirúrgicos donde la mayoría de los encuestados: 39,7% están de acuerdo, 46%, estos grupos de encuestados interpretan de forma enfática los beneficios que pueden tener la utilidad de la realidad virtual, aunque reconocen que no reemplaza totalmente los métodos tradicionales de la enseñanza médica. Reseña también, que un menor porcentaje difiere enfáticamente que pueda influir de manera positiva el aprendizaje, que podría deberse a mala praxis, experiencias previas muy negativas en el uso práctico de la realidad virtual o por limitaciones metodológicas y/o tecnológicas.

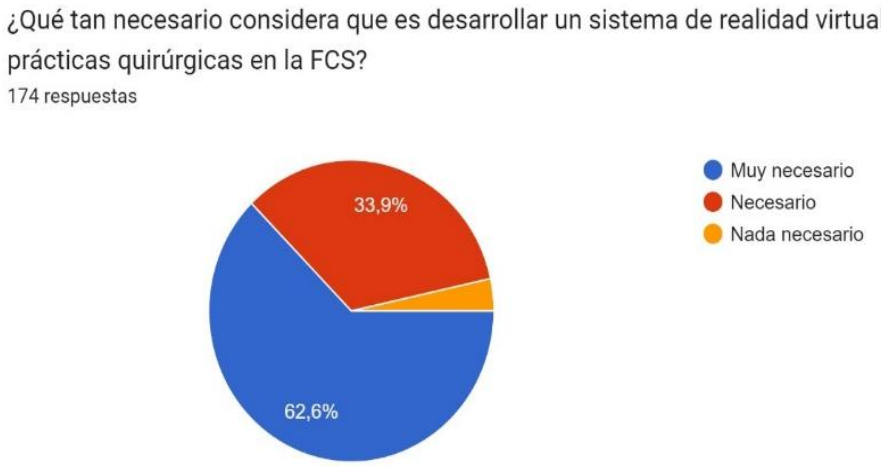


Figura 8: Conocimiento Previo sobre Realidad Virtual

En la Figura 8, los encuestados valoraron positivamente en un 62,6%, que los sistemas virtualizados deben ser desarrollados para practicas quirúrgicas en entornos virtuales, dando un gran consenso de la aceptación sobre el potencial en ofrecer calidad en el aprendizaje médico dentro del ámbito de la FCS. Sin embargo, hay un pequeño grupo que considera innecesario la utilidad de estas plataformas, sugiriendo escepticismo o desconfianza ya sea a poca información de los avances innovadores de estas tecnologías o preferencia por los procedimientos de enseñanza tradicionales.



Figura 9: Opiniones sobre la Eficacia de la Realidad Virtual en la Educación

En la Figura 9, una mayoría de los encuestados creen enfáticamente en los sistemas virtualizados para soluciones quirúrgicas, pero no de manera que pueda sustituir los métodos tradicionales de aprendizaje en la enseñanza médica, se puede interpretar como sugerencia tecnológica para integrarse junto a los procedimientos tradicionales de enseñanza y/o minimizar las limitaciones que puedan tener algunos laboratorios físicos. Es importante resaltar que un 14, 4 % está en desacuerdo como solución viable para sustituir los laboratorios sugiriendo a la falta de información o seguir con los métodos tradicionales de enseñanza.

En la Figura 10, en la gran mayoría de los encuestados con un porcentaje muy significativo siente empatía con el uso de los sistemas virtuales como medio tecnológico al integrase con la enseñanza médica tradicional, es decir un 48,3% totalmente de acuerdo y 40, 2% de acuerdo. Esto conlleva a analizar que la mayoría de los estudiantes de las ciencias de la salud pudieran adaptarse sin restricciones de ningún tipo en el uso de estas plataformas virtuales, debido a la familiaridad con estas

herramientas en el contexto actual. Solo un pequeño porcentaje del grupo está en desacuerdo o totalmente desacuerdo, pudiéndose interpretar la falta de accesibilidad, dificultad o desconfianza en el uso de la realidad virtual como método de enseñanza.

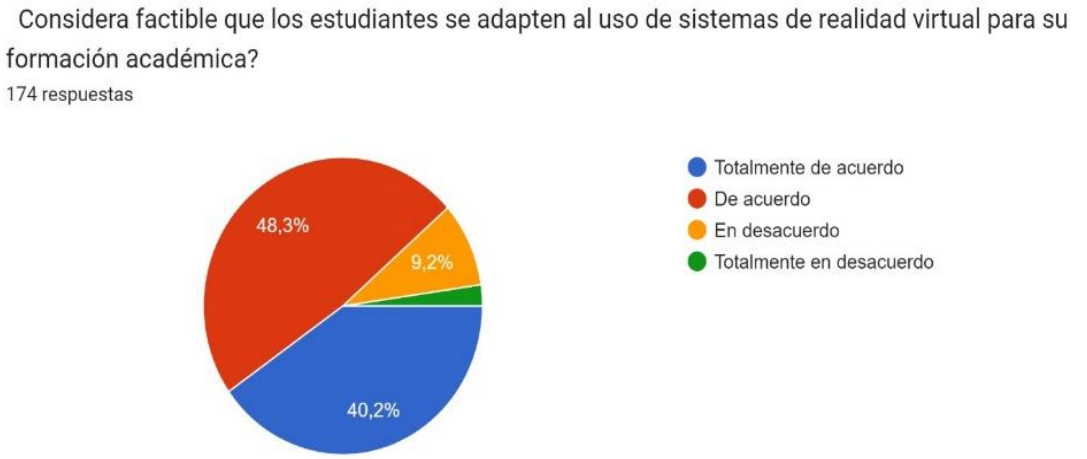


Figura 10: Barreras Percibidas para el Uso de Realidad Virtual en su formación académica



Figura 11: Interés en Capacitación sobre Tecnologías Emergentes

En la Figura 11, con 51,7% de acuerdo y 34,5% totalmente de acuerdo los encuestados manifestaron un gran interés en la realidad virtual para facilitar las prácticas fuera del laboratorio físico. Esto puede deberse a las innovaciones tecnológicas ya desarrolladas tienen mayor calidad en los gráficos que simula un mejor realismo de las prácticas y una eficiencia en el acceso al laboratorio lo que puede redundar en la calidad de la educación médica o crear expectativas para su implementación. Un pequeño grupo

manifestó su desacuerdo o totalmente desacuerdo con tener las prácticas fuera del laboratorio debido a las faltas de confianza o de información en este método.



Figura 12: Frecuencia de Participación en Capacitación Tecnológica

En la Figura 12, en esta interpretación del gráfico se refleja un elevado consenso (muy útil o útil: (92,5 %) en cual los beneficios de los sistemas virtualizados tienen un gran potencial en la educación médica, para superar obstáculos y restricciones por el acceso al laboratorio físico. Los encuestados pueden interpretar esta pregunta como una implementación complementaria o sustituto total para las praxis del laboratorio, aunque una minoría (7,5 %) no creen o no tienen confianza en estas plataformas como sustitución de prácticas quirúrgicas y que pueden aún más obteniendo información para mejorar la percepción o efectividad de estas plataformas, mejorando sistemas de alta calidad gráfica, para en ofrecer un mayor realismo o acceso a la tecnología de manera general.



Figura 13: Efecto de la Pandemia en el Uso de Herramientas Tecnológicas

En la Figura 13, describe que existe una fuerte opinión positiva hacia el desarrollo de un proyecto de realidad virtual que podría crear mejoras en la distribución del tiempo de práctica en comparación con los procedimientos de enseñanzas tradicionales con un resultado 55,2% estar de acuerdo que la RV tiene beneficios para dar tiempos de prácticas de manera eficiente, un tercio de los encuestados tiene una opinión positiva mucho más fuertes sobre estos beneficios potenciales de la RV y representa el 88% del grupo encuestado. El 9,8% del grupo muy reducido está en desacuerdo, lo que refuerza la tendencia positiva de la implementación de los sistemas virtualizados. El resultado aquí reflejado puede deberse a que los sistemas virtualizados son más flexibles a la hora de obtener praxis personalizadas y estarían de acuerdo con las necesidades particulares de cada, algo que no es permitido por los procedimientos normales de educación médica.

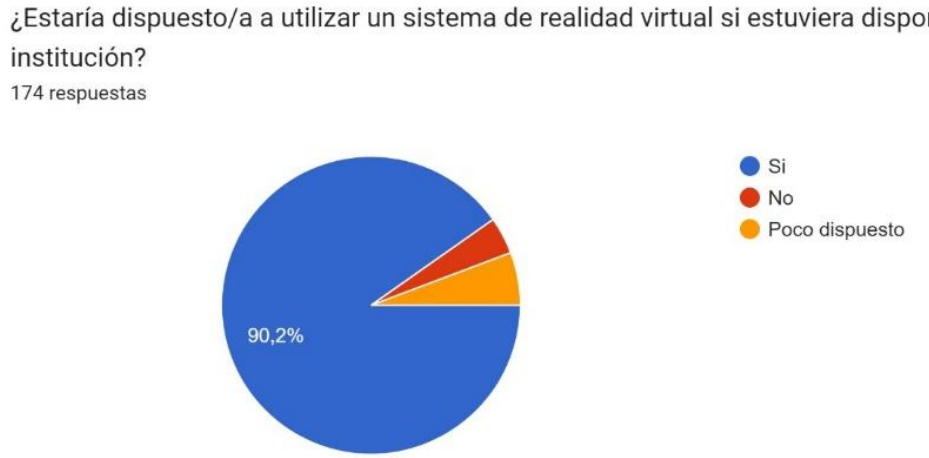


Figura 14: Preferencias de Formato de Capacitación (Virtual, Presencial, Híbrido)

En la Figura 14, aquí la tendencia favorable sobre la percepción que tienen los encuestados sobre la RV es incuestionable. Esto puede sugerir que la RV es una tecnología innovadora en pleno avance en su desarrollo para ser implementada e integrada junto con los métodos tradicionales de aprendizaje médico. Puede existir una conexión o interés motivado a las ventajas y beneficios que ofrece en el área educativa en general para el mejoramiento de las prácticas de laboratorio.



Figura 15: Impacto de la Realidad Virtual en las Habilidades Prácticas

En la Figura 15, un gran porcentaje sigue la tendencia positiva de una opinión generalizada hacia los sistemas virtualizados con un 47,7% y 45,4 % como totalmente de acuerdo y de acuerdo, sugiriendo que estos sistemas permitirán una educación médica de calidad en la FCS como herramienta complementaria al ser integrada a la enseñanza quirúrgica. El grupo que no está de acuerdo (minoritario) puede deberse a la falta de familiaridad o con poca información y conocimiento de estas plataformas tecnológicas.

En la Figura 16, en esta pregunta refleja la importancia que tiene las opiniones de los estudiantes en el área de la salud en colaboración de expertos y estudiantes más avanzados que puedan ofrecer opiniones, sugerencias y expectativas sobre los conocimientos ya adquirido en virtud de mejorar eficientemente la formación académica con prácticas y permitir que el desarrollo e implementación de un sistema virtualizado sea eficiente y efectivo en la enseñanza-aprendizaje de la FCS sean ajustadas a la realidad académica.

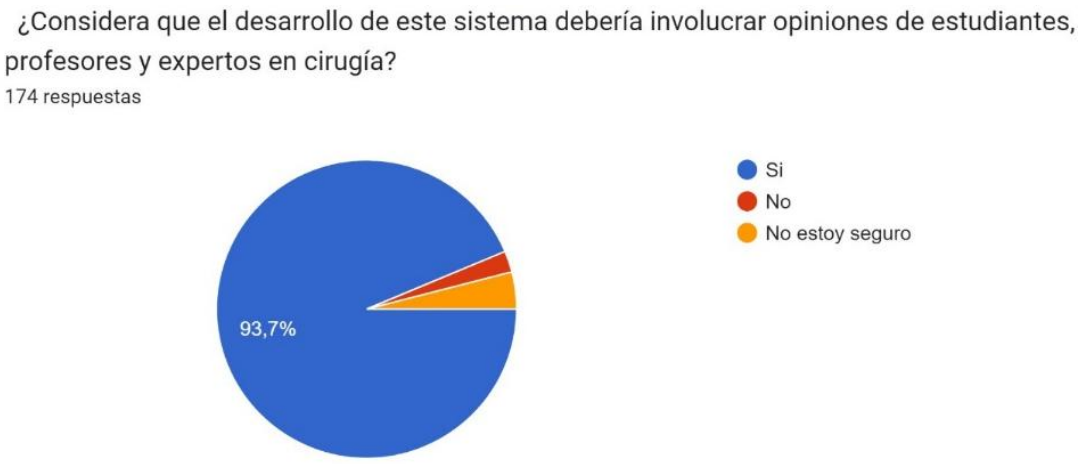


Figura 16: El Futuro del Aprendizaje y las Tecnologías Emergentes

La encuesta constó de 16 preguntas en total, sin embargo, la primera y última son para romper el hielo y facilitar la comunicación con los encuestados. Las demás preguntas se enfocaron sobre el tema de estudio.

d. Aplicación de las Entrevistas

El software ATLAS. Ti es una potente herramienta para el análisis cualitativo de datos, ampliamente utilizada en la investigación para la gestión, codificación y teorización de grandes volúmenes de información textual, gráfica, de audio y video. En el contexto de la entrevista realizada por la licenciada en enfermería, ATLAS.ti sirvió como la plataforma principal para la organización y el análisis de la transcripción. Permitió la identificación sistemática de temas emergentes, la codificación de segmentos relevantes del texto que evidenciaron las percepciones, experiencias y opiniones de la entrevistada en relación con el tema de estudio [63].

En la Figura 17, se muestra la interfaz del software ATLAS.ti, donde se evidencia el proceso de análisis cualitativo de la entrevista realizada al experto en desarrollo. En la sección central, se visualiza el texto de la entrevista, segmentado y potencialmente codificado con etiquetas que representan temas o conceptos clave identificados en el discurso del experto. A la derecha, se observa el "Explorador de citas", donde se listan los segmentos codificados, permitiendo al investigador organizar y revisar las evidencias textuales relevantes para cada categoría analítica. La aplicación de códigos

aplicados al texto demuestra el uso de ATLAS.ti como una herramienta para estructurar y analizar de manera sistemática la información obtenida, facilitando así la identificación de patrones, relaciones para una comprensión profunda de la perspectiva del experto.

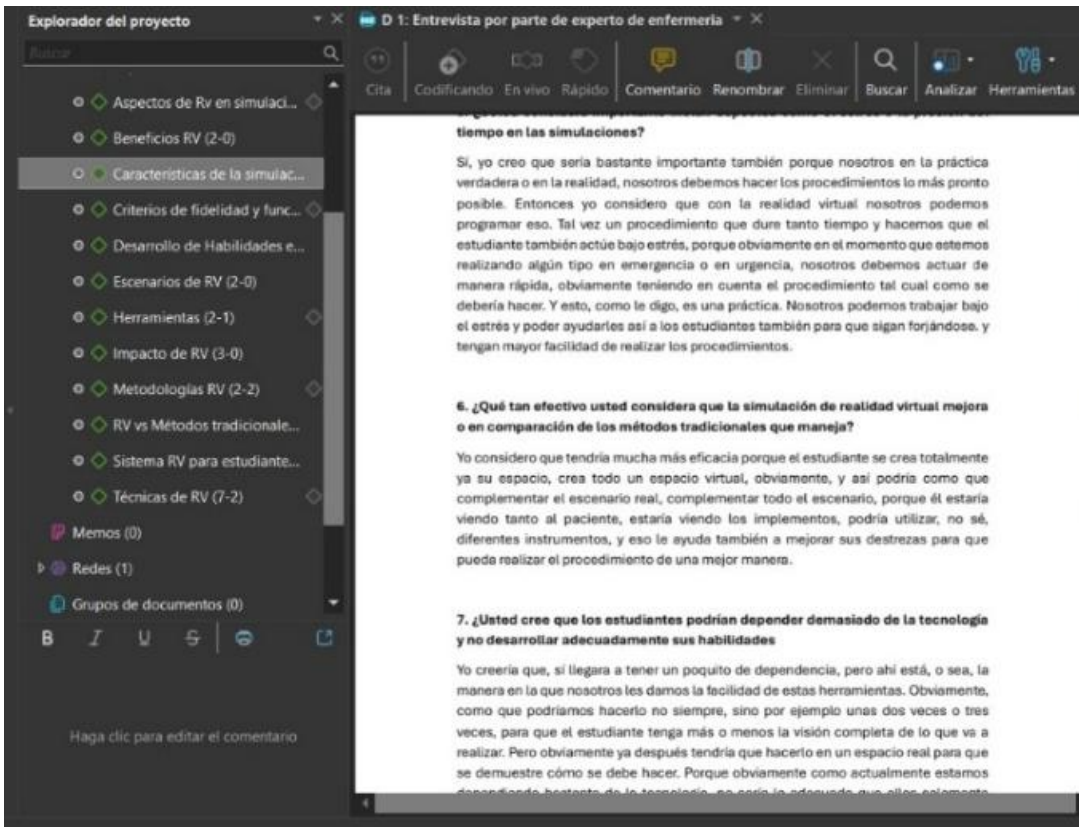


Figura 17:Evidencia del Análisis en ATLAS.ti de la Entrevista al Experto en Desarrollo

La Figura 18 se muestra, el análisis cualitativo de la entrevista con la licenciada en enfermería. Se visualiza el texto de la entrevista, indicando la aplicación de temáticas relevantes para la investigación.

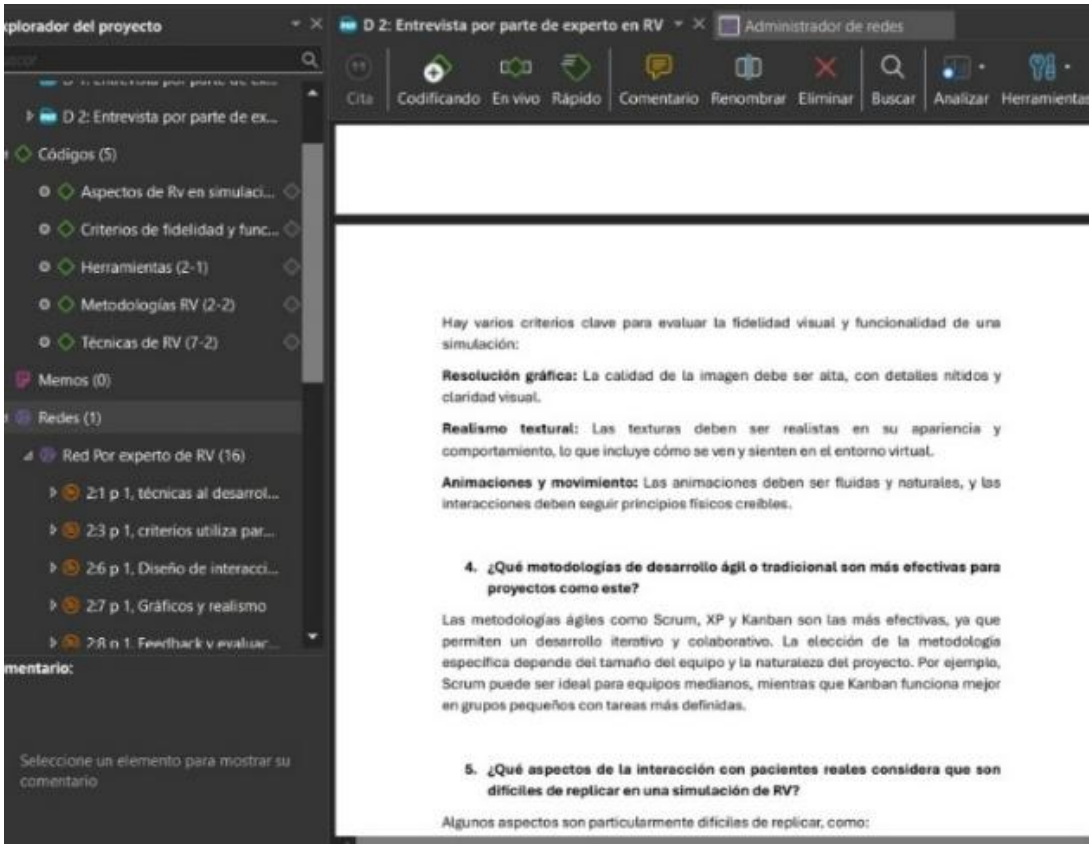


Figura 18:Evidencia del Análisis en ATLAS.ti de la Entrevista de la Licenciada en Enfermería

e. Nube de Palabras

Atlas.ti permite visualizar una nube de palabras con los conceptos más representativos o las palabras con mayor repetición o frecuencia del discurso del informante lo que repercute en una codificación más acertada. El tamaño de cada vocablo en la visualización es directamente proporcional a su frecuencia de aparición en la transcripción. De esta manera, las palabras de mayor tamaño indican los temas centrales y recurrentes en el discurso del experto, sugiriendo los focos principales de la conversación. Mientras que, las palabras de menor tamaño, aunque menos frecuentes, aún contribuyen al contexto general de las opiniones y conocimientos [63].

En la Figura 19, la nube de palabras generada a partir de la entrevista con el experto en desarrollo visualiza la frecuencia de los términos empleados. Este patrón sugiere que la conversación se enfocó principalmente en aspectos como ["simulador virtual", "desarrollo", "interacción", “lenguajes de programación”], revelando las prioridades

del experto en este campo. Las palabras de menor tamaño, aunque menos prominentes, complementan el contexto del dialogo, señalando conceptos secundarios o detalles relevantes desde su perspectiva en el desarrollo de simuladores.

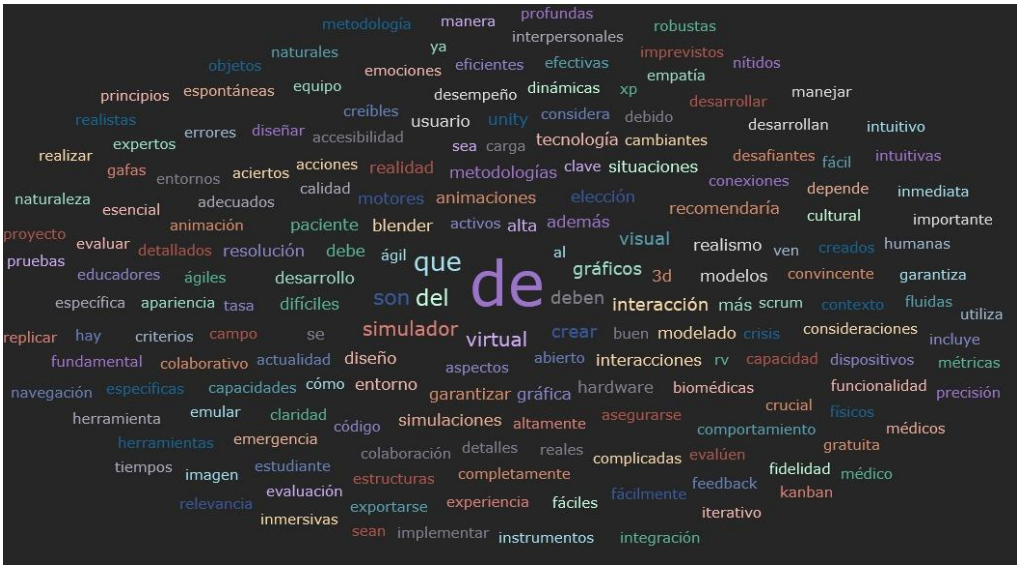


Figura 19:Nube de Palabras de la Entrevista al Experto en Desarrollo



Figura 20:Nube de Palabras de la Entrevista a la Licenciada en Enfermería

f. Redes – Networks

En Atlas.ti aparte de generar nubes de palabras, también nos ayuda a crear redes, permiten representar gráficamente las conexiones entre diferentes conceptos presentes

en el discurso de los informantes. Estas conexiones son el resultado de las relaciones de coocurrencia presente entre los códigos y facilita la reflexión del investigador. La red conceptual generada nos ayuda de manera significativa a representar y explorar sus estructuras conceptuales individuales y, potencialmente, a identificar puntos de convergencia o divergencia entre sus perspectivas [63].

El análisis de las redes conceptuales derivadas de las entrevistas revela enfoques diferentes según la experiencia de los entrevistados. En la Figura 21, se presenta la red conceptual de la entrevista al experto en desarrollo, centrada en "Técnicas de RV" y "Metodología RV", revela que el dialogo abarcó aspectos clave como colaboración, usabilidad, hardware, feedback, gráficos, diseño de interacción y criterios de fidelidad y funcionalidad (animación, realismo, resolución) en la simulación. Estos nodos y sus conexiones ilustran la estructura del discurso del experto y los temas centrales que articuló en relación con el desarrollo de la realidad virtual.

Por su parte, la Figura 22 presenta la red conceptual generada a partir de la entrevista con la licenciada en enfermería, cuyo eje central es el "Sistema RV para estudiantes". De este nodo principal emergen cuatro dimensiones analíticas:

- Escenarios de RV (con especial atención a su parecido con situaciones reales)
- Aspectos de RV en estudiantes" (incluyendo estrés, tiempo en simulaciones, y análisis costo-beneficio)
- Impacto de RV" en los procesos de aprendizaje
- Comparación directa entre RV vs Métodos tradicionales médicos

Esta configuración demuestra cómo la profesional concibe la implementación de tecnologías inmersivas en formación médica, estableciendo relaciones entre los entornos virtuales, sus efectos pedagógicos y su contraste con las prácticas convencionales.

Tras el análisis de las entrevistas con ATLAS.ti, se identificaron elementos clave para el desarrollo del sistema. Se destacó la importancia de interfaces intuitivas, gráficos realistas, retroalimentación inmediata y el uso de herramientas como Unity y Blender.

A nivel educativo, se sugirió incluir escenarios realistas con estrés y presión de tiempo. También se reconocieron beneficios como la práctica segura y repetible, el aprendizaje inmersivo y la evaluación de habilidades, junto con desafíos como la emulación de aspectos humanos y la necesidad de complementar la RV con prácticas reales.

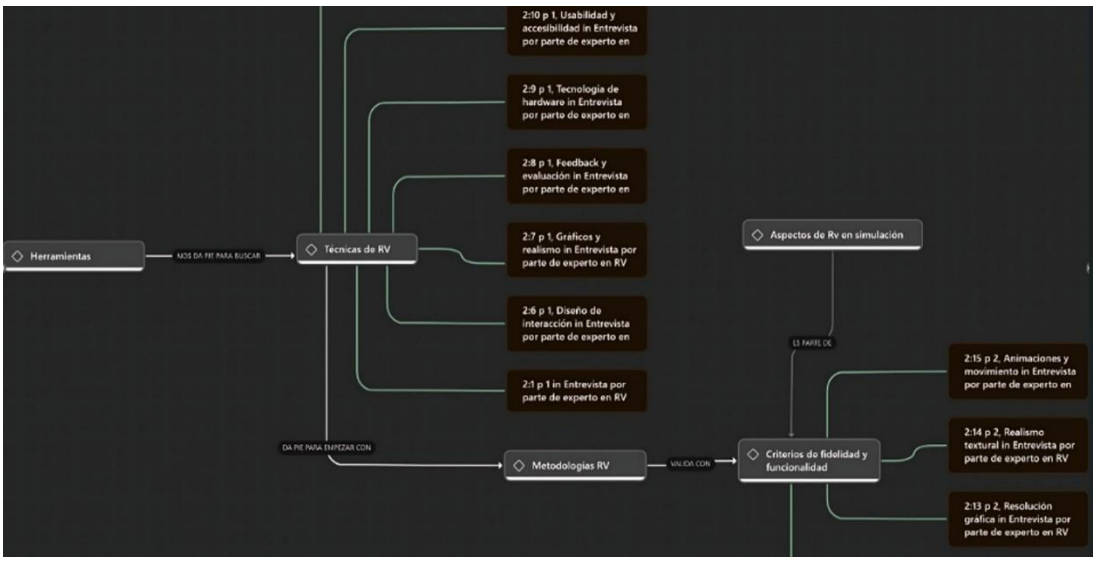


Figura 21:Representación de la Estructuras Conceptuales: Entrevista al Experto en Desarrollo

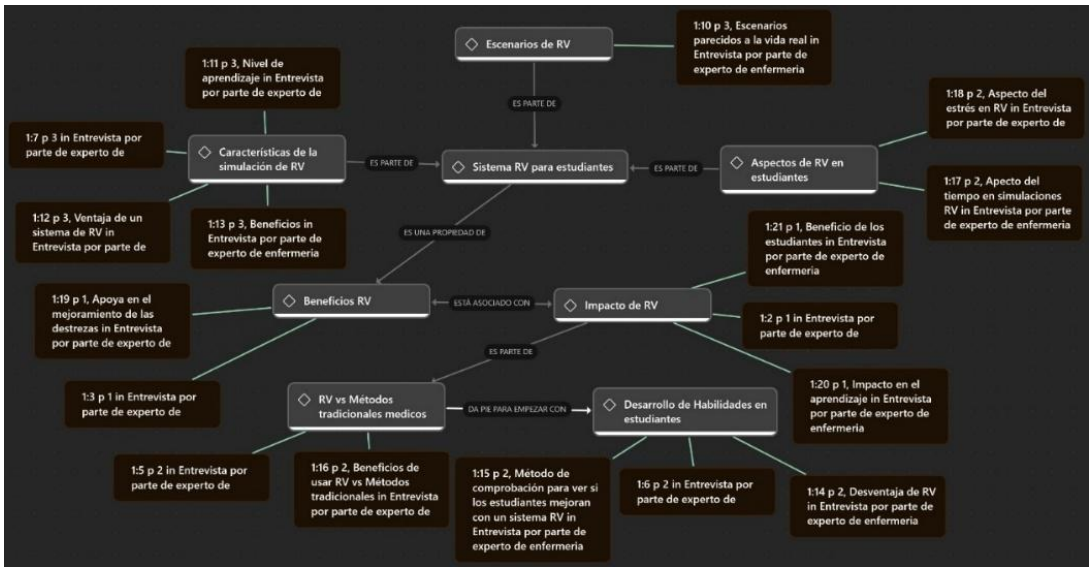


Figura 22:Representación de la Estructuras Conceptuales: Entrevista a la Licenciada en Enfermería

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

El presente capítulo expone los resultados obtenidos a partir de la selección y análisis de las herramientas tecnológicas empleadas en el desarrollo del sistema de realidad virtual propuesto. En el Capítulo 1 **(**Fundamentación teórica**)** se establecieron los fundamentos conceptuales que sustentan este proyecto, incluyendo los principales motores gráficos, lenguajes de programación, herramientas de modelado 3D y dispositivos de visualización inmersiva, con énfasis en su aplicación en contextos educativos del área de la salud.

Además, se desarrollará una metodología para medir la complejidad técnica, la cual se basó en un conjunto de métricas normalizadas (Complejidad en el Desarrollo de Entornos de Realidad Virtual para Educación en Enfermería, Métricas para medir complejidad de desarrollo, Métricas de Rendimiento en Tiempo Real, Métricas de Experiencia de Usuario (UX)).

Por otro lado, se detalla la adopción de la metodología XP (Extreme Programming) como enfoque ágil para el desarrollo del sistema. Esta metodología fue seleccionada por su capacidad de adaptarse a entornos académicos, su enfoque iterativo, su flexibilidad ante cambios y su énfasis en la entrega continua de funcionalidades. La combinación entre una evaluación técnica rigurosa y una estrategia de desarrollo ágil permite garantizar que el sistema cumpla con los objetivos definidos, sea funcional en dispositivos como Meta Quest 3 y se adapte al contexto educativo y operativo en el que será implementado.

3.1 Tablas comparativas de Motores, Lenguajes, Gafas de RV y Herramientas de Modelado

Una vez analizadas y observadas las diferencias de cada una de las herramientas utilizadas para crear sistemas de realidad virtual, se han obtenido los siguientes resultados:

Tabla 1. Motores de desarrollo VR

Motor	Lenguaje	Características clave	Licencia	Uso destacado
Unity	C# (.NET)	Interacciones inmersivas, multiplataforma	Gratuita y de Paga	Videojuegos, simulación
Unreal	C++	Visualización en tiempo real, gráficos avanzados	Gratuita	Simulación
Godot	GDScript, C#, C++	Código abierto, sistema de viewports, multiplataforma	Gratuita	Juegos 2D/3D, educación
CryEngine	C++	Alto realismo gráfico, uso intensivo de hardware	Gratuita (con límites)	Simulación

Se seleccionó Unity como motor de desarrollo ya que como se muestra en la Tabla 1 su integración directa con C#, su capacidad para crear experiencias inmersivas de alta calidad y su compatibilidad multiplataforma son ideales para este proyecto. Además, cuenta con una comunidad activa y abundante documentación, lo que facilita el desarrollo. Esta elección también se alinea con los requisitos técnicos establecidos previamente para el proyecto.

En la Tabla 2 se observa los diferentes lenguaje de programación el cual se escogió C#, por su facilidad de uso, su enfoque en programación orientada a objetos (POO) y su integración nativa con Unity . Esto permite una implementación eficiente y coherente con el entorno de desarrollo adoptado.

Tabla 2. Lenguajes de programación comunes en VR

Lenguaje	Características clave	Motores relacionados	Nivel
C#	POO, fácil de usar, multiplataforma	Unity, Godot	Alto
C++	Eficiente, compleja, para proyectos grandes	Unreal, CryEngine	Muy alto
Java	Portabilidad (JVM)	General, Cardboard	Alto
Python	Sintaxis clara, interpretado, versátil	Prototipos, herramientas	Medio
JavaScript	WebVR, integración con Three.js	WebVR, Three.js	Medio
Swift	Apple, integración AR/VR, eficiente	iOS apps, ARKit	Alto

Tabla 3. Comparación de Gafas de realidad virtual

Modelo	Resolución	Tecnología destacada	Tecnología destacada
Meta Quest 3s	2064 x 2208	Realidad mixta, tasa hasta 120Hz	VR/AR general
HTC Vive Pro	2880 x 1600	AMOLED, audio mejorado	Profesional/gaming
Valve Index	1440 x 1600	Cámaras RGB, SteamVR	Juegos inmersivos
PlayStation VR2	2000 x 2040	OLED, audio 3D	PS5 juegos VR

La Tabla 3 muestra porque se eligieron las Meta Quest 3s, ya que, su excelente equilibrio entre resolución, soporte de realidad mixta, alta tasa de refresco y compatibilidad con Unity la vuelven las ideales para este proyecto. Además, su portabilidad y rendimiento las hacen adecuadas para simulaciones inmersivas educativas.

Tabla 4. Herramientas de modelado 3D

Software	Tipo de licencia	Características clave	Uso
Blender	Open-source	Rigging, animación, renderizado, comunidad activa	General, juegos, cine
Autodesk Maya	Licencia comercial	Estándar industrial, animación compleja	Juegos AAA, cine, VFX
Cinema 4D	Licencia comercial	Flujo de trabajo accesible, diseño gráfico, AR/VR	Visualización, VFX

Se optó por Blender ya que como se detalla en la Tabla 4, su naturaleza de código abierto, su amplia funcionalidad profesional (modelado, animación, rigging y renderizado), y su compatibilidad directa con Unity son óptimos para este proyecto. Unity Es una herramienta flexible y accesible que permite optimizar los recursos del proyecto sin sacrificar calidad.

Tras un análisis comparativo de las características técnicas, compatibilidad de lenguajes y ventajas competitivas de los principales motores de desarrollo, se ha seleccionado Unity como plataforma base para la implementación del sistema. Esta determinación se fundamenta en:

- Su integración nativa con C# como lenguaje principal de programación
- Su capacidad demostrada para crear experiencias inmersivas de alta calidad
- El soporte de una comunidad global activa y documentación extensa
- Su arquitectura multiplataforma que garantiza compatibilidad con diversos dispositivos

La elección cumple con los requisitos técnicos establecidos en los lineamientos del proyecto, donde se especificó el uso de Unity como estándar de desarrollo. Para la implementación final, se utilizarán: Dispositivo de visualización: Meta Quest 3, seleccionado por sus capacidades de realidad mixta y rendimiento óptimo y para el

modelado Blender, elegido por su código abierto, versatilidad profesional y flujo de trabajo compatible con Unity.

Una vez realizada la documentación previa y escogido las herramientas que se van a usar sobre los lenguajes de programación, motores de desarrollo y herramientas de modelado 3D, se llevó a cabo una investigación más profunda donde se analizaron diferentes estudios y aplicaciones reales de sistemas de realidad virtual en contextos quirúrgicos. En la Tabla 5, resume los resultados cuantitativos obtenidos de diversas investigaciones aplicadas al uso de sistemas de realidad virtual en entornos quirúrgicos y educativos. Cada fila corresponde a un estudio en el que se evaluaron métricas específicas como el tiempo de tarea, la precisión en la identificación del instrumental, la comprensión conceptual, la confianza del usuario o la satisfacción general. Los escenarios analizados abarcan procedimientos como la apendicectomía laparoscópica, la cesárea, la cirugía torácica y la formación perioperatoria en enfermería. Los resultados reflejan mejoras significativas en el desempeño de los participantes al utilizar simulaciones inmersivas, con reducciones claras en los tiempos de ejecución y aumentos relevantes en puntuaciones de evaluación, confianza y percepción de realismo. Asimismo, el meta-análisis citado proporciona evidencia estadística robusta sobre el impacto positivo de estas tecnologías en el aprendizaje, respaldando la eficacia de la realidad virtual como herramienta formativa en contextos médicos.

La Tabla 6 resume las características técnicas de las herramientas de desarrollo empleadas en diversos estudios enfocados en la simulación quirúrgica mediante realidad virtual. Se detalla el lenguaje de programación utilizado, el motor gráfico o plataforma de desarrollo y el tipo de visor o dispositivo HMD aplicado en cada investigación. Es notable la prevalencia del motor Unity, tanto en su versión estándar como combinado con dispositivos hápticos, lo que refuerza su versatilidad en entornos educativos inmersivos. Asimismo, lenguajes como C y C++ son predominantes, lo que refleja su compatibilidad con simuladores de alto rendimiento. En cuanto a los visores, las gafas Oculus Rift y HTC Vive aparecen con frecuencia, lo que demuestra su aceptación por parte de la comunidad científica en experiencias de entrenamiento quirúrgico. Esta información permite justificar, desde un enfoque comparativo, la elección de herramientas en el desarrollo del sistema propuesto para la identificación y armado del instrumental quirúrgico.

Tabla 5: Tabla Comparativa de Herramientas de Realidad Virtual en Cirugía

Herramienta / Contexto	**Escenario procedimental**	**Métrica evaluada**	**Resultado cuantitativo**	**Fuente**
Motor VR (simulación laparoscópica)	Apendicetomía laparoscópica	Tiempo de tarea	Inexpertos: 864.5s; Intermedios: 477.2s; Expertos: 352.1s (p¡0.01)	[64], [65]
Simulaciones VR Cesárea (HMD)	Técnica de Cesárea y RPM	Mini-test y confianza	Puntuación: VR 42 vs Control 36 (p¡0.001); Confianza ↑ (p = 0.014)	[65]
Sistema VR para enfermería	Identificación y preparación instrumental	Realismo y comprensión	Puntuación en realismo y comprensión significativamente mayor en VR (p ¡ 0.05)	[66]
Entrenamiento con VR inmersivo (RCT cesárea)	Cesárea y RPM	Mini-test y confianza	Mediana test VR 42 vs 36 (p¡0.001); mayor confianza post-VR	[67]
Ensayo clínico enfermería torácica (VR)	Cirugía torácica	Puntuación evaluación	Mediana 520 vs 440 puntos (p=0.04); satisfacción 8.5/10	[68]
Desarrollo contenido VR + háptica	Educación perioperatoria	Usabilidad y presencia	Alta presencia, usabilidad y satisfacción en 29 estudiantes	[69]
Meta-análisis ortopedia (23 RCTs)	Educación ortopédica general	SMD conocimiento/operación	SMD conocimiento 1.08, operación 1.44, diseño 1.75 (p¡0.001)	[70]

Tabla 6: Tecnologías Utilizadas por los estudios

Estudio / Caso	Lenguaje de Programación	Motor de Desarrollo	Gafas
Motor VR (simulación laparoscópica)	C++ / C	Unity / VIST VR	Óculos Rift
Simulaciones VR Cesárea (HMD)	C	Unity 3D	HTC Vive / Óculos Quest
Sistema VR para enfermería	C	Unity	Cardboard / Óculos
Entrenamiento con VR inmersivo (RCT cesárea)	C	Unity	HTC Vive / Meta Quest
Ensayo clínico enfermería torácica (VR)	C	Unity	Óculos Rift
Desarrollo contenido VR + háptica	C++ / C	Unity + Dispositivo háptico	HTC Vive + guantes hápticos
Meta-análisis ortopedia (23 RCTs)	C++ / C (varios)	Unity / Unreal / VRTK	Múltiples: Óculos, Vive, Hololens

Luego del análisis de los casos de aplicación de la realidad virtual en entornos quirúrgicos y de la identificación de las tecnologías a utilizar, se consideró pertinente incorporar una evaluación comparativa de carácter técnico. Con este fin, se elaboró un estudio que propone un índice de complejidad herramientas de desarrollo de VR, basado en métricas como el tiempo de aprendizaje, los requisitos del sistema, la facilidad de implementación, la calidad de la documentación y la curva de productividad. Esta evaluación permitió valorar de forma objetiva el grado de dificultad asociado al uso de cada herramienta, considerando tanto sus prestaciones como las barreras de entrada que representan para equipos de desarrollo académico o institucional.

A partir de esta evaluación, se clasificaron herramientas como Unity, Unreal Engine, Blender y Autodesk Maya, revelando diferencias significativas en cuanto a accesibilidad y eficiencia.

3.2 Evaluación Complejidad herramientas de desarrollo de VR

Se desarrolló un índice de complejidad multidimensional basado en cinco métricas principales:

- **Tiempo de Aprendizaje (TA):** Horas requeridas para competencia básica
- **Requisitos del Sistema (RS):** Especificaciones hardware mínimas
- **Facilidad de Implementación (FI):** Tiempo para primera aplicación funcional
- **Documentación y Soporte (DS):** Calidad de recursos de aprendizaje
- **Curva de Productividad (CP):** Tiempo para alcanzar productividad profesional

a. Fórmula del Índice de Complejidad

$$\mathrm{IC} = (\mathrm{TA} * 0.25) + (\mathrm{RS} \times 0.20) + (\mathrm{FI} \times 0.20) + (\mathrm{DS} \times 0.15) + (\mathrm{CP} \times 0.20)$$

Donde cada componente se normaliza en una escala de 1-10 (1 = menor complejidad, 10 = mayor complejidad).

b. Componentes y Pesos Explicados:

TA (Tiempo de Aprendizaje) - Peso: 25%

- **Qué mide:** Horas necesarias para alcanzar competencia básica
- **Por qué es importante:** Es el factor más crítico para la adopción inicial
- **Escalamiento:**
    - 1-2 = 20-40 horas (muy fácil)
    - 3-4 = 40-80 horas (fácil)
    - 5-6 = 80-120 horas (moderado)

- 7-8 = 120-200 horas (difícil)
- 9-10 = >200 horas (muy difícil)

RS (Requisitos del Sistema) - Peso: 20%

- **Qué mide:** Demandas de hardware/software
- **Escalamiento:**
  - 1-3 = Requisitos bajos (4GB RAM, GPU básica)
  - 4-6 = Requisitos moderados (8GB RAM, GPU dedicada)
  - 7-10 = Requisitos altos (16GB+ RAM, GPU high-end)

FI (Facilidad de Implementación) - Peso: 20%

- **Qué mide:** Tiempo para crear primera aplicación funcional
- **Escalamiento:**
  - 1-3 = 1-5 horas para primer prototipo
  - 4-6 = 5-20 horas
  - 7-10 = >20 horas

DS (Documentación y Soporte) - Peso: 15%

- **Qué mide:** Calidad de recursos de aprendizaje disponibles
- **Criterios:**
  - Documentación oficial
  - Tutoriales disponibles
  - Comunidad activa
  - Soporte técnico

CP (Curva de Productividad) - Peso: 20%

- **Qué mide:** Tiempo para alcanzar productividad profesional
- **Diferencia con TA:** TA mide competencia básica, CP mide dominio avanzado
- **Justificación de los Pesos:**
    - TA (25%): Mayor peso porque determina la barrera de entrada
    - RS y FI (20% c/u): Impacto directo en costos y tiempo de desarrollo
    - CP (20%): Crucial para proyectos comerciales a largo plazo
    - DS (15%): Menor peso porque es factor de apoyo, no determinante

c. Criterios de Evaluación

La evaluación se basó en:

- Revisión de literatura académica especializada
- Análisis de documentación oficial
- Estudios de caso de proyectos VR documentados
- Métricas de adopción en la industria

d. Resultados

Motores Gráficos

Unity 3D

La Tabla 7, presentada resume los resultados del análisis de complejidad aplicado al motor Unity 3D, evaluado mediante un conjunto de métricas estandarizadas. Se observa que el tiempo estimado para alcanzar una competencia básica en esta herramienta se encuentra entre 40 y 60 horas, lo cual le otorga una puntuación baja

(3.0) en la escala de complejidad. Asimismo, sus requisitos del sistema son moderados (3.5), y la facilidad de implementación se considera alta (2.5), lo que indica que permite lograr resultados funcionales en poco tiempo. En cuanto a la documentación y soporte, Unity destaca con una calificación excelente (2.0), gracias a su amplia comunidad y recursos oficiales bien estructurados. Su curva de productividad se clasifica como gradual (4.0), permitiendo a los desarrolladores mejorar su rendimiento de forma progresiva. Todo esto se traduce en un índice de complejidad total de 3.2, clasificado como bajo, lo que lo convierte en una opción ideal para proyectos académicos y prototipos de simulación. Unity es ampliamente favorecido por su interfaz amigable y extensa tienda de activos, haciéndolo accesible para desarrolladores de todos los niveles. Su soporte multiplataforma permite desarrollar una vez y desplegar en varios dispositivos VR [71].

Tabla 7: Análisis de Desempeño Técnico de Unity 3D

Métrica	**Valor**	**Puntuación (1-10)**
Tiempo de Aprendizaje	40-60 horas	3.0
Requisitos del Sistema	Moderados	3.5
Facilidad de Implementación	Alta	2.5
Documentación y Soporte	Excelente	2.0
Curva de Productividad	Gradual	4.0
Índice de Complejidad	3.2	Baja

$$IC = (3.0 \times 0.25) + (3.5 \times 0.20) + (2.5 \times 0.20) + (2.0 \times 0.15) + (4.0 \times 0.20)$$

$$IC = 0.75 + 0.70 + 0.50 + 0.30 + 0.80 = 3.05 \approx 3.2$$

Unreal Engine

Tabla 8: Evaluación de la Complejidad Técnica del Motor Unreal Engine

Métrica	**Valor**	**Puntuación (1-10)**
Tiempo de Aprendizaje	80-120 horas	6.5
Requisitos del Sistema	Altos	8.0
Facilidad de Implementación	Moderada	7.0
Documentación y Soporte	Buena	6.0
Curva de Productividad	Pronunciada	8.5
Índice de Complejidad	7.1	Alta

En la Tabla 8, muestra la evaluación técnica del motor de desarrollo Unreal Engine, según un conjunto de métricas que permiten determinar su nivel de complejidad. El tiempo de aprendizaje estimado para adquirir un manejo básico oscila entre 80 y 120 horas, lo que le otorga una puntuación alta (6.5). Además, presenta requisitos de sistema elevados (8.0), lo que implica la necesidad de equipos de alto rendimiento para ejecutar el entorno de desarrollo de forma fluida. La facilidad de implementación se considera moderada (7.0), mientras que la documentación y el soporte son catalogados como buenos (6.0). En cuanto a la curva de productividad, esta es pronunciada (8.5), lo que indica una mayor inversión de tiempo hasta alcanzar un nivel profesional. El índice global de complejidad resultante es de 7.1, clasificado como alto, lo cual sugiere que, si bien Unreal ofrece un alto potencial gráfico, no es la opción más accesible para desarrolladores con recursos limitados.

Se destaca además que Unreal puede resultar muy exigente para la mayoría de las laptops estándar, mientras que motores como Unity pueden ejecutarse adecuadamente incluso en equipos de gama media con varios años de uso. Esta diferencia en requisitos técnicos puede ser determinante a la hora de seleccionar una plataforma de desarrollo para proyectos educativos o prototipos académicos [72].

$$IC = (6.5 \times 0.25) + (8.0 \times 0.20) + (7.0 \times 0.20) + (6.0 \times 0.15) + (8.5 \times 0.20)$$

$$IC = 1.625 + 1.60 + 1.40 + 0.90 + 1.70 = 7.225 \approx 7.1$$

Software de Modelado 3D

Blender

Tabla 9: Evaluación de la Complejidad Técnica del Software Blender

Métrica	**Valor**	**Puntuación (1-10)**
Tiempo de Aprendizaje	60-100 horas	6.0
Requisitos del Sistema	Moderados	4.0
Facilidad de Implementación	Moderada	6.5
Documentación y Soporte	Buena	7.0
Curva de Productividad	Pronunciada	8.5
Índice de Complejidad	6.8	Moderada-Alta

La Tabla 9, expone el análisis de complejidad del software de modelado 3D Blender, evaluado según criterios técnicos clave. El tiempo estimado de aprendizaje se ubica entre 60 y 100 horas, con una puntuación de 6.0, lo que indica una curva de inicio moderadamente exigente. Sus requisitos del sistema son considerados moderados (4.0), mientras que la facilidad de implementación recibe una calificación de 6.5, lo cual sugiere una experiencia técnica aceptable, pero con ciertos desafíos iniciales. La documentación y el soporte obtienen una buena calificación (7.0), gracias a la amplia comunidad de usuarios y recursos disponibles en línea. Por otro lado, la curva de productividad se considera pronunciada (8.5), señalando que se requiere un periodo prolongado para alcanzar fluidez profesional. En conjunto, Blender alcanza un índice de complejidad de 6.8, ubicado en la categoría "moderada-alta".

Cabe destacar que, al tratarse de un software gratuito y de código abierto, Blender ofrece una interfaz compleja pero altamente personalizable, lo cual representa tanto una ventaja como un reto para nuevos usuarios. Su flexibilidad lo convierte en una herramienta potente para proyectos de modelado 3D, aunque su dominio demanda tiempo y constancia [73].

$$IC = (6.0 \times 0.25) + (4.0 \times 0.20) + (6.5 \times 0.20) + (7.0 \times 0.15) + (8.5 \times 0.20)$$

$$IC = 1.50 + 0.80 + 1.30 + 1.05 + 1.70 = 6.35 \approx 6.8$$

Autodesk Maya

Tabla 10: Evaluación de la Complejidad Técnica del Software Autodesk Maya

Métrica	Valor	Puntuación (1-10)
Tiempo de Aprendizaje	100-150 horas	8.0
Requisitos del Sistema	Altos	7.5
Facilidad de Implementación	Baja	8.5
Documentación y Soporte	Excelente	3.0
Curva de Productividad	Muy Pronunciada	9.0
Índice de Complejidad	8.2	Alta

La Tabla 10, muestra la evaluación técnica del software Autodesk Maya, una herramienta profesional de modelado 3D ampliamente utilizada en la industria del

cine, televisión y videojuegos. El tiempo requerido para adquirir una competencia funcional se sitúa entre 100 y 150 horas, con una puntuación de 8.0, lo que evidencia una curva de entrada elevada. Sus requisitos del sistema también son altos (7.5), y la facilidad de implementación es baja (8.5), indicando que el entorno presenta barreras iniciales significativas. A pesar de ello, cuenta con documentación excelente (3.0), y su curva de productividad se considera muy pronunciada (9.0), lo que significa que se necesita una inversión considerable de tiempo para alcanzar un nivel productivo avanzado. El índice de complejidad general asciende a 8.2, clasificándolo como una herramienta de alta complejidad.

Autodesk Maya se posiciona como un estándar en la creación de activos 3D de alta calidad, siendo la elección habitual en producciones audiovisuales de gran escala. A pesar de su complejidad, su potencia, precisión y conjunto de funcionalidades lo han consolidado como una de las herramientas más completas y exigentes del mercado [74].

$$IC = (8.0 \times 0.25) + (7.5 \times 0.20) + (8.5 \times 0.20) + (3.0 \times 0.15) + (9.0 \times 0.20)$$

$$IC = 2.00 + 1.50 + 1.70 + 0.45 + 1.80 = 7.45 \approx 8.2$$

e. Análisis Estadístico

Análisis Estadístico

Tabla 11: Distribución Comparativa del Índice de Complejidad (IC)

Herramienta	**IC**	**Categoría**
Unity	3.2	Baja
Blender	6.8	Moderada-Alta
Unreal Engine	7.1	Alta
Maya	8.2	Alta

- **Media:** 6.325
- **Desviación Estándar:** 2.18
- **Coeficiente de Variación:** 34.5%

Como se observa en la Tabla 11, presenta una comparación directa del Índice de Complejidad (IC) entre distintas herramientas utilizadas en el desarrollo de experiencias de realidad virtual y modelado tridimensional. Unity destaca con el valor más bajo (3.2), clasificado dentro de la categoría de complejidad baja, lo que confirma su accesibilidad tanto para principiantes como para desarrolladores con recursos limitados. Blender, en cambio, presenta una complejidad moderada-alta (6.8), resultado de su potencia combinada con una interfaz de aprendizaje gradual. Unreal Engine y Autodesk Maya ocupan las posiciones más exigentes, con índices de 7.1 y 8.2 respectivamente, ambos dentro de la categoría de complejidad alta. Esto sugiere que, si bien ofrecen capacidades avanzadas, también requieren mayor inversión en tiempo de aprendizaje, recursos técnicos y experiencia previa.

Correlación entre Métricas

El análisis de correlación entre las métricas revela relaciones significativas que permiten comprender mejor los factores que influyen en la complejidad de las herramientas de desarrollo.

La correlación positiva alta entre el tiempo de aprendizaje y la curva de productividad ($r = 0.89$, $p < 0.01$) indica que cuanto más tiempo se requiere para aprender una herramienta, más prolongado será el periodo necesario para alcanzar un nivel productivo profesional.

Por otro lado, la relación negativa entre los requisitos del sistema y la facilidad de implementación ($r = -0.76$, $p < 0.05$) sugiere que, a mayor demanda de recursos técnicos, menor es la facilidad para poner en marcha una aplicación funcional de manera ágil.

Finalmente, se observa una correlación negativa entre la calidad de la documentación y el índice de complejidad ($r = -0.72$, $p < 0.05$), lo que implica que mejores recursos

de aprendizaje y soporte contribuyen a reducir la percepción de dificultad general. Estos hallazgos aportan una base cuantitativa sólida para respaldar decisiones en la selección de herramientas tecnológicas para entornos educativos o de prototipado.

3.3 Análisis de requerimientos

Una vez finalizada la fase de recopilación de información mediante encuestas y entrevistas, se procedió a la formulación de los requerimientos del sistema. Los datos obtenidos de las encuestas resultaron fundamentales para establecer los requerimientos funcionales, permitiendo especificar las capacidades y operaciones que el sistema debía incorporar. Por otro lado, el análisis de las entrevistas realizadas a expertos proporcionó la base para definir los requerimientos no funcionales, relacionados con atributos de calidad como rendimiento, usabilidad y seguridad del sistema.

Un requisito, o también denominado requerimiento, es definido por el glosario de la IEEE como una condición o capacidad que debe satisfacer o poseer un sistema o un componente de un sistema para satisfacer un contrato, un estándar, una especificación. Los requerimientos se clasifican en dos categorías principales: requisitos funcionales (RF), que especifican las funciones y comportamientos del sistema, y requisitos no funcionales (RNF), que establecen criterios de calidad y restricciones de operación [75].

3.3.1 Requerimientos Funcionales

Los requerimientos funcionales describen las funciones específicas que debe cumplir un sistema, incluyendo sus entradas, salidas y el manejo de excepciones. Estos requerimientos dependen del tipo de software que se desarrolla, del perfil de los usuarios y de los objetivos que la organización desea alcanzar con el sistema. Pueden ir desde especificaciones generales, que definen las capacidades básicas del sistema, hasta requerimientos detallados que responden a necesidades particulares de la organización, como prácticas locales o la integración con sistemas existentes[76].

Los requerimientos funcionales, identificados mediante entrevistas a expertos y usuarios clave, definen las acciones y capacidades específicas que el sistema debe

ofrecer para cumplir con sus objetivos educativos. Se centran en las funcionalidades principales que permitirán la simulación quirúrgica efectiva y personalizada, asegurando la interacción del usuario con el entorno virtual de manera guiada, evaluativa y progresiva.

3.3.2 Requerimientos no Funcionales

Los requerimientos no funcionales, recopilados mediante encuestas, se enfocan en aspectos de calidad que influyen directamente en la experiencia del usuario, la sostenibilidad del sistema y su proyección a futuro. Estos requisitos garantizan que el sistema sea amigable, accesible, eficiente y capaz de evolucionar conforme a las necesidades del entorno educativo y tecnológico. Su adecuada implementación asegura no solo una experiencia de uso óptima, sino también un desarrollo técnico sólido y escalable [75].

El análisis de los requerimientos funcionales y no funcionales se fundamenta en la norma IEEE 29148:2018, ya que este estándar proporciona directrices actualizadas y completas para la especificación y gestión de requisitos en sistemas y software. Dicha norma establece un enfoque estructurado que permite describir de manera clara y precisa tanto las funcionalidades del sistema (requerimientos funcionales) como los atributos de calidad (requerimientos no funcionales), tales como el rendimiento, la seguridad y la usabilidad. Al aplicar la IEEE 29148:2018, se favorece la consistencia, completitud, verificabilidad y trazabilidad de los requisitos, lo cual contribuye a la reducción de riesgos durante el ciclo de vida del desarrollo. Además, esta norma promueve buenas prácticas de ingeniería de software, asegurando que el producto final cumpla con las expectativas y necesidades del cliente [77].

3.3.3 Especificación de Requerimientos de Software (SRS) basado en la norma ISO 29148:2018

PROYECTO: “SISTEMA DE REALIDAD VIRTUAL DEDICADO A LA IDENTIFICACIÓN Y ARMADO DEL INSTRUMENTAL QUIRÚRGICO PARA ESTUDIANTES DE ENFERMERÍA”

a. Introducción

Propósito

El propósito es definir los requerimientos funcionales y no funcionales para el SISTEMA DE REALIDAD VIRTUAL DEDICADO A LA IDENTIFICACIÓN Y ARMADO DEL INSTRUMENTAL QUIRÚRGICO PARA ESTUDIANTES DE ENFERMERÍA.

Alcance del sistema

El sistema de realidad virtual permitirá tanto a estudiantes como docentes visualizar la sala de operaciones (quirófano), realizar la simulación de armado del instrumentar quirúrgico en los procedimientos de cesárea y apendicetomía.

Definiciones, acrónimos y abreviaturas

- RA: Realidad Aumentada
- SRS: Software Requirements Specification

Referencias

- ISO/IEC/IEEE 29148:2018
- Sistema de realidad virtual, simulación de quirófano para el armado de la mesa mayo

Visión general

El Sistema de Realidad Virtual para la Identificación y Armado del Instrumental Quirúrgico está diseñado para transformar la formación de los estudiantes de enfermería mediante una herramienta tecnológica inmersiva. Tiene como finalidad desarrollar una plataforma en realidad virtual que simule un quirófano en entornos 3D, permitiendo a los usuarios interactuar con el instrumental quirúrgico (fotografías) utilizado en procedimientos de cesáreas y apendicetomías.

El sistema busca identificar el reconocimiento adecuado de los instrumentos, al mismo tiempo que funciona como un recurso educativo complementario para los docentes, optimizando la enseñanza mediante simulaciones virtuales seguras y libres de riesgos. Su implementación contribuirá a minimizar errores en el manejo del instrumental en situaciones reales.

La plataforma se desarrollará bajo estándares de calidad educativa y tecnológica, asegurando una experiencia de usuario intuitiva, realista. Se pretende modernizar la metodología de enseñanza en el ámbito de la salud, ofreciendo una solución innovadora que combine educación y tecnología.

b. Descripción General

Perspectiva del Producto

El producto será un sistema de realidad virtual para que los estudiantes de enfermería aprendan a identificar el instrumental quirúrgico mediante simulaciones inmersivas de cesáreas y apendicetomías. Reemplaza prácticas tradicionales, ofreciendo un entorno seguro, interactivo y repetible que reduce los riesgos.

Funcionalidades del Producto

- Visualización del escenario quirúrgico.
- Interacción con el avatar.
- Interacción con el menú de operación.
- Interacción con el menú del instrumental.
- Retroalimentación automática.

Características de los Usuarios

Tabla 12: Características de los usuarios a usar el sistema

Tipo de Usuario	Descripción	Nivel Técnico
Estudiantes	Acceder a la simulación del armado del instrumental quirúrgico	Alto

En la Tabla 12, se presentan las características del tipo de usuario principal de la aplicación de realidad virtual: los estudiantes. Este grupo está orientado a interactuar con la simulación del armado del instrumental quirúrgico y se considera con un nivel técnico alto, ya que se espera que tengan conocimientos previos en el área médica y estén familiarizados con el uso de tecnologías educativas avanzadas.

Restricciones

- El sistema debe ejecutarse en dispositivos Meta Quest 3.
- Lenguaje de desarrollo: C#.
- Motor de videojuego: Unity.
- Modelado del inmueble: Blender.

Supuestos y dependencias

- Se debe contar con conexión a internet donde se va a realizar las pruebas.
- Los estudiantes deben ser capacitados para movilizarse e interactuar con la simulación.

c. Requerimientos Específicos

Requerimientos Funcionales

- **RF01:** Pantalla de inicio, el sistema debe mostrar una pantalla de inicio donde tendrá el nombre de la Facultad de Ciencia de la Salud y dos botones: "Iniciar" y "Salir". Al presionar "Salir", se debe cerrar la aplicación. Al presionar "Iniciar", se debe cambiar a la escena del quirófano.

- **RF02**: Creación y personalización del avatar, se desarrollará un avatar el cual debe ser un médico con su respectivo traje al momento de entrar a un quirófano.

- **RF03:** Navegación del avatar, el avatar debe poder caminar, girar y retroceder libremente dentro del quirófano virtual.

- **RF04:** Navegación del avatar, el sistema debe permitir seleccionar entre dos tipos de operación: cesárea y apendicetomía.

- **RF05:** Selección de tiempos quirúrgicos, el sistema debe permitir seleccionar uno de los cuatro tiempos quirúrgicos según el tipo de operación escogido.

- **RF06:** Selección de instrumentos quirúrgicos, el sistema debe permitir seleccionar los instrumentos quirúrgicos correspondientes al tipo de operación y al tiempo quirúrgico.

- **RF07:** Armado de mesa mayo, los instrumentos quirúrgicos seleccionados correctamente se deben colocar automáticamente en la mesa mayo. Si se selecciona un instrumento incorrecto, el sistema debe mostrar un mensaje de advertencia indicando que no corresponde.

- **RF08:** Crear menú de tipo de operación, el sistema debe mostrar un menú en el cual el usuario pueda seleccionar el tipo de operación (cesárea o apendicetomía) y, según la selección, mostrar los tiempos quirúrgicos correspondientes.

- **RF09:** Crear menú de instrumentos quirúrgicos, el sistema debe mostrar un menú con todos los instrumentos quirúrgicos disponibles. Este menú debe permitir al usuario seleccionar los instrumentos necesarios para el tipo de operación y el tiempo quirúrgico previamente seleccionados.

- **RF10:** Creación del escenario quirúrgico, el sistema debe generar un escenario de quirófano realista que incluya todos los elementos propios de una sala quirúrgica, como camilla, lámparas, mesa mayo, mobiliario médico, entre otros. Este escenario debe permitir la navegación del avatar en primera persona.

- **RF11:** Creación del instrumental, para la creación del instrumental se usará solo imágenes en formato SVG, dado que no se despixela al aumentar o disminuir su tamaño.

Requerimientos No Funcionales

Tabla 13: Requerimientos No Funcionales

ID	Requerimiento	Categoría
RNF01	El sistema debe ser completamente compatible y ejecutable en gafas de realidad virtual Meta Quest 3	Compatibilidad / Portabilidad
RNF02	El quirófano virtual debe tener modelos realistas de acuerdo con el entorno de la vida real, para asegurar una experiencia inmersiva de entrenamiento.	Usabilidad / Experiencia de usuario
RNF03	Las acciones del usuario, como la selección de operaciones, tiempos e instrumentos deben tener una respuesta del sistema menor a 5 segundo.	Rendimiento
RNF04	El sistema debe proporcionar retroalimentación al momento de tomar un instrumento que no pertenezca a un respectivo tiempo.	Usabilidad / Interacción del usuario
RNF05	La interfaz debe ser intuitiva, con controles fáciles de aprender y usar, especialmente para usuarios sin experiencia previa en realidad virtual.	Usabilidad / Accesibilidad

En la Tabla 13 se detallan los requerimientos no funcionales de la aplicación de realidad virtual, los cuales son fundamentales para garantizar una experiencia de usuario efectiva y fluida. Entre estos se incluye la compatibilidad con dispositivos Meta Quest 3, el realismo visual del quirófano virtual, la capacidad de respuesta del sistema ante acciones del usuario, la retroalimentación interactiva y una interfaz intuitiva. Estos aspectos aseguran que la aplicación sea accesible, inmersiva y eficiente para los estudiantes durante el proceso de aprendizaje práctico.

d. Requerimientos de interfaces externas

Interfaz de usuario

- Diseño inmersivo, adaptado al campo visual y la interacción natural en gafas Meta Quest 3.
- Interacción mediante seguimiento de manos y controladores VR, permitiendo seleccionar operaciones, tiempos quirúrgicos e instrumentos.
- Menús contextuales en 2D, adaptados al entorno quirúrgico simulado.
- Retroalimentación visual al seleccionar objetos, cometer errores o completar acciones correctas.
- Interfaz organizada por tiempos quirúrgicos, guiando al usuario según el flujo de la cirugía seleccionada.

Interfaz de hardware

- Compatibilidad total con gafas de realidad virtual Meta Quest 3.

Interfaz de software

- El motor de videojuegos Unity debe operar la versión 2022.3.18.f1.

Atributos del sistema

- El sistema debe poder ser empaquetado y desplegado como una aplicación nativa para Meta Quest 3, utilizando entornos compatibles con Android/Unity.
- Debe estar diseñado para que el código sea modular, escalable y mantenible, siguiendo principios de arquitectura limpia, facilitando la futura inclusión de nuevas cirugías o herramientas.
- El sistema debe garantizar una experiencia fluida y sin interrupciones.

3.4 Métrica Cyclomatic Complexity (complejidad de desarrollo)

La Cyclomatic Complexity es una métrica utilizada para medir la complejidad del código fuente de un programa. Cuantifica el número de rutas linealmente independientes a través de un programa, contando cada punto de decisión que puede afectar el flujo de ejecución. Basada en la teoría de grafos, representa el programa como un grafo de flujo de control. Una Cyclomatic Complexity alta indica un código complejo y menos mantenible, mientras que un valor bajo sugiere un código más simple y fácil de entender. Esta métrica ayuda a identificar riesgos en el mantenimiento y las pruebas, guiando hacia un software más eficiente, fiable y mantenible [78].

Los puntos de decisión incluyen: declaraciones if, cláusulas else, estructuras switch, bucles for y while, operadores lógicos && y ||, bloques catch y operadores ternarios. Es importante tener en cuenta que una mayor Cyclomatic Complexity afecta la mantenibilidad del código, requiere pruebas más exhaustivas, se correlaciona con una mayor probabilidad de errores y puede tener implicaciones en el rendimiento [79].

a. ***SonarQube***

SonarQube es una herramienta de código abierto que permite verificar de forma continua la calidad del código fuente. Se encarga de analizar el código para identificar problemas como duplicación, malas prácticas, brechas en la cobertura de pruebas, errores y vulnerabilidades. Además, genera informes detallados que facilitan la comprensión de estos aspectos. SonarQube es compatible con múltiples lenguajes de programación, incluyendo Java, C#, JavaScript, Python, TypeScript y Kotlin. Entre sus características principales se encuentra la capacidad de identificar errores antes de que lleguen a producción, destacar problemas de diseño y de codificación, detectar vulnerabilidades y riesgos de seguridad, y monitorear la cobertura de pruebas unitarias mediante su integración con otras herramientas. También permite a los equipos establecer estándares y políticas de codificación para mantener la calidad a lo largo del desarrollo [80].

Tabla 14: Resumen general de funcionalidades y uso de una herramienta de análisis de calidad de código

Aspecto	Resumen
Características	Análisis de calidad, soporte multilenguaje, detección de errores, CI/CD
Ventajas	Interfaz intuitiva, extensible, buena integración, comunidad activa
Desventajas	Alto consumo de recursos, configuración compleja, funciones premium
Casos ideales	Equipos medianos-grandes, proyectos con CI/CD y enfoque en seguridad
Requisitos técnicos	Equipos medianos-grandes, proyectos con CI/CD y enfoque en seguridad

La Tabla 14, muestra un resumen general de las características principales, ventajas, desventajas, casos de uso recomendados y requisitos técnicos de una herramienta de análisis de calidad de código. Este recurso facilita la comprensión de los aspectos más relevantes que deben considerarse para su aplicación en entornos de desarrollo de software.

*b. **Visual Studio Analyzer***

Visual Studio Analyzer es una herramienta integrada en Visual Studio que se utiliza para analizar el rendimiento de las aplicaciones durante su ejecución. Su función principal es registrar información detallada sobre el comportamiento de la aplicación en aspectos como eficiencia, velocidad de ejecución y flujo de control. Esto permite a los desarrolladores identificar cuellos de botella, comportamientos inesperados y oportunidades de optimización. Además, ofrece información general del sistema, así como datos específicos sobre grupos de funciones o métodos individuales, proporcionando así una visión profunda del funcionamiento interno de la aplicación en tiempo real [81].

Tabla 15: Resumen de Visual Studio Analyzer: características, ventajas y aplicación técnica

Aspecto	Resumen
Características	Análisis de rendimiento, flujos de ejecución, monitoreo de base de datos
Ventajas	Integración con Visual Studio, fácil uso, retroalimentación inmediata
Desventajas	Solo para Microsoft, posible impacto en rendimiento, funciones limitadas
Casos ideales	Optimización de apps .NET, análisis de CPU, memoria y bases de datos
Requisitos técnicos	Usa ETW, instrumentación dinámica, requiere entorno Windows

La Tabla 15, presenta un resumen de las funciones clave, beneficios, limitaciones y requerimientos técnicos de Visual Studio Analyzer. Esta herramienta, integrada al entorno de desarrollo de Microsoft, está orientada al análisis de rendimiento en aplicaciones .NET, facilitando la detección de cuellos de botella y la optimización de recursos en tiempo de ejecución.

*c. **Mapa comparativo***

Tabla 16: Mapa Comparativo SonarQube vs Visual Studio Analyzer

Aspecto	SonarQube	Visual Studio Analyzer
Tipo	Análisis calidad continua	Análisis de rendimiento
Enfoque principal	Calidad código	Rendimiento ejecución
Lenguajes	+20 lenguajes	Principalmente .NET
Despliegue	Servidor independiente	Integrado en IDE
Análisis	Estático	Dinámico
Precio base	Edición Community gratuita	Incluido en VS
Interfaz	Web	Dentro de Visual Studio
Plataforma	Multiplataforma	Windows
Reportes	Dashboards web personalizables	Gráficos y tablas en IDE
Medición rendimiento	Limitada	Avanzada
Comunidad	Grande, activa	Ecosistema Microsoft

- **Selección de herramienta y aplicación sobre el código del proyecto**

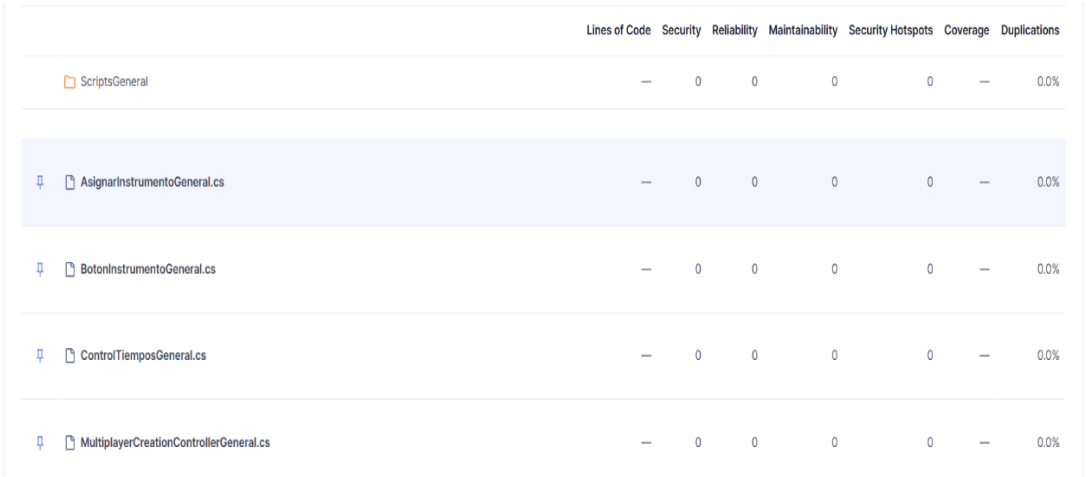
Tras la exhaustiva evaluación de las herramientas de análisis de código (ver Tabla 16), se procedió a implementar SonarQube como solución para la métrica Cyclomatic Complexity. A continuación, se presentan los hallazgos y métricas obtenidos durante el análisis del código fuente, que permitieron identificar áreas críticas de mejora y establecer una línea base para la calidad del proyecto.



	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
Test-Complejidad-NurseTools	21,304	2	343	112	1	0.0%	12.9%
Assets	17,111	2	343	112	1	0.0%	13.4%
UIElementsSchema	4,193	0	0	0	0	—	0.0%

Figura 23: Análisis de Calidad de Código Dashboard de SonarQube

Al analizar el sistema en SonarQube, se identificaron varias métricas clave que reflejan el estado del código (ver Figura 23). En esta imagen se muestra una tabla detallada del análisis SonarQube del proyecto "Test-Complejidad-NurseTools" junto con sus subdirectorios. El proyecto completo consta de 21,304 líneas de código, presenta 2 problemas de seguridad, 343 problemas de confiabilidad, 112 problemas de mantenibilidad, 1 punto crítico de seguridad, 0.0% de cobertura de pruebas y 12.9% de duplicaciones. Desglosa los resultados por directorios principales, mostrando que el directorio "Assets" contiene la mayoría de los problemas (17,111 líneas de código con idénticos 2 problemas de seguridad, 343 de confiabilidad y 112 de mantenibilidad, y un 13.4% de duplicaciones), mientras que "UIElementsSchema" (con 4,193 líneas) no presenta ningún problema en ninguna categoría y tiene 0% de duplicaciones. Esto señala claramente que los problemas detectados en el proyecto se concentran específicamente en el directorio "Assets".



	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
ScriptsGeneral	—	0	0	0	0	—	0.0%
AsignarInstrumentoGeneral.cs	—	0	0	0	0	—	0.0%
BotonInstrumentoGeneral.cs	—	0	0	0	0	—	0.0%
ControlTiemposGeneral.cs	—	0	0	0	0	—	0.0%
MultiplayerCreationControllerGeneral.cs	—	0	0	0	0	—	0.0%

Figura 24:Análisis de Calidad de Código del Directorio Scripts - Métricas SonarQube

En la Figura 24, se muestra una tabla de análisis de SonarQube correspondiente al directorio "Scripts", el cual contiene todos los archivos de programación del proyecto. Esta sección específica presenta métricas de calidad de código para varios archivos .cs (C#) relacionados con la funcionalidad general del sistema. La tabla refleja resultados perfectos para todos los archivos analizados, con cero problemas detectados en todas las categorías evaluadas (seguridad, confiabilidad, mantenibilidad y puntos críticos de seguridad), así como un 0.0 % de duplicación de código. A partir de estos resultados, se concluye que los problemas detectados no provienen de la programación del sistema en sí, sino de los archivos de configuración generados por defecto al crear el proyecto (como dependencias, extensiones y otros componentes automáticos).

La Figura 25, muestra la sección de "Issues" de SonarQube, específicamente de los problemas de seguridad detectados en el proyecto. El problema actual se titula "Verify the origin of the received message”. Este problema de seguridad está relacionado con la verificación del origen de mensajes en comunicaciones cross-origin en JavaScript. SonarQube identifica que el código actual no verifica el origen de los mensajes recibidos antes de procesarlos. Para resolver este problema, se aplicarán los cambios que nos sugiere SonarQube. Esto se podría lograr añadiendo una validación de origen en la función subscribe que aparece en el código, verificando que el evento provenga de una fuente confiable antes de ejecutar.

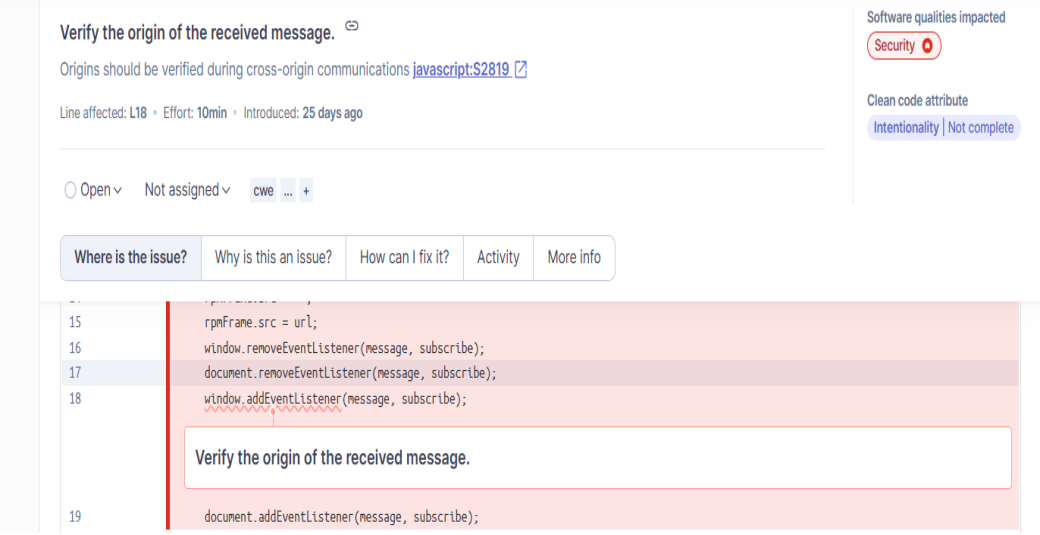


Figura 25: Análisis de Vulnerabilidad de Seguridad

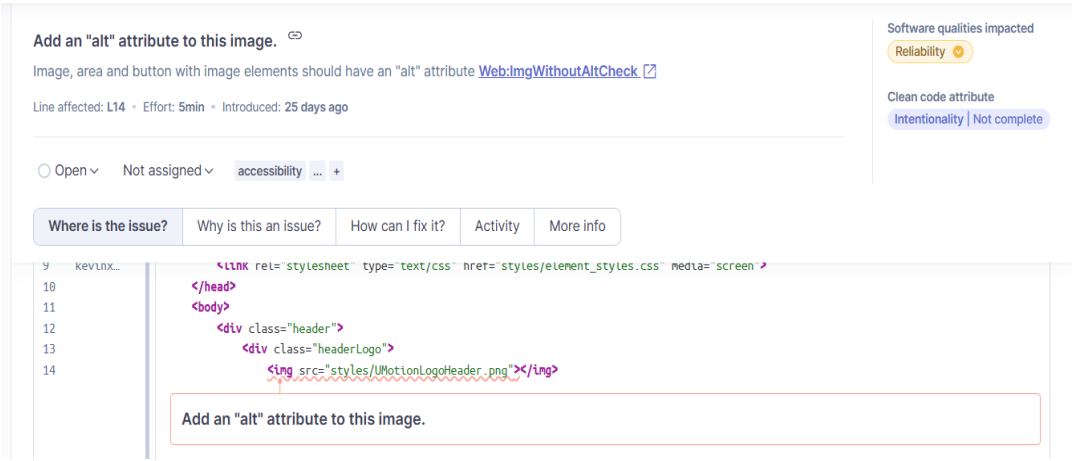


Figura 26:Problemas de confiabilidad detectados por SonarQube

En la Figura 26, se evidencia un conjunto de advertencias generadas por SonarQube relacionadas con la falta del atributo "alt" en elementos <img> dentro del código HTML del sistema. Este atributo es esencial para mejorar la accesibilidad y confiabilidad del sitio, ya que permite describir el contenido de las imágenes en contextos donde no pueden ser visualizadas. La herramienta indica un total de 343 incidencias de este tipo, las cuales afectan directamente la calidad del software en cuanto a confiabilidad. Estos problemas serán corregidos siguiendo las recomendaciones específicas del propio SonarQube, que sugieren añadir descripciones significativas en el atributo "alt" para garantizar un código más limpio, accesible y conforme a buenas prácticas de desarrollo web.

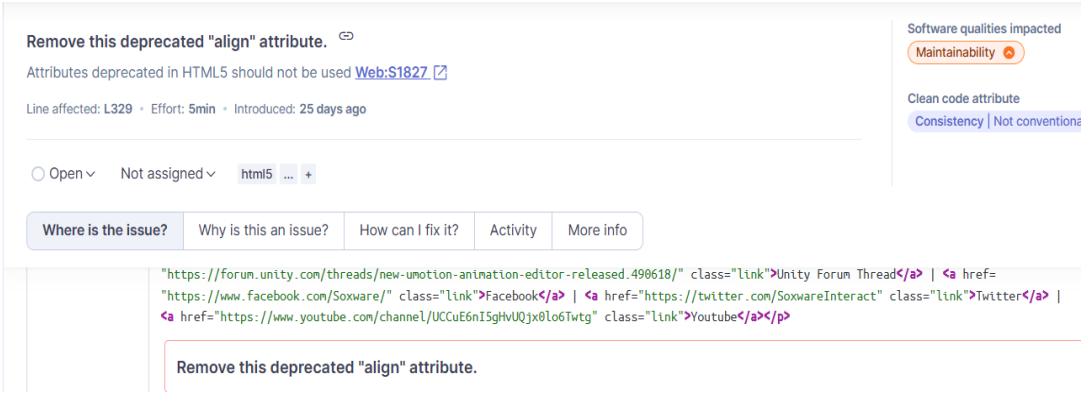


Figura 27: Problemas de mantenibilidad detectados por SonarQube debido al uso del atributo obsoleto

En la Figura 27, se muestran advertencias emitidas por SonarQube relacionadas con el uso del atributo HTML obsoleto "align" en varios archivos del sistema. Este atributo ha sido desaprobado en HTML5 y su uso afecta negativamente la mantenibilidad del código, ya que introduce inconsistencias con los estándares modernos. En total, se identifican 112 incidencias de este tipo, las cuales afectan la consistencia del código y dificultan su mantenimiento a largo plazo. Para resolver estos inconvenientes, se aplicarán las recomendaciones proporcionadas por SonarQube, las cuales sugieren eliminar dicho atributo y utilizar estilos CSS actualizados, lo que permitirá mejorar la calidad estructural del sistema y asegurar su compatibilidad futura.

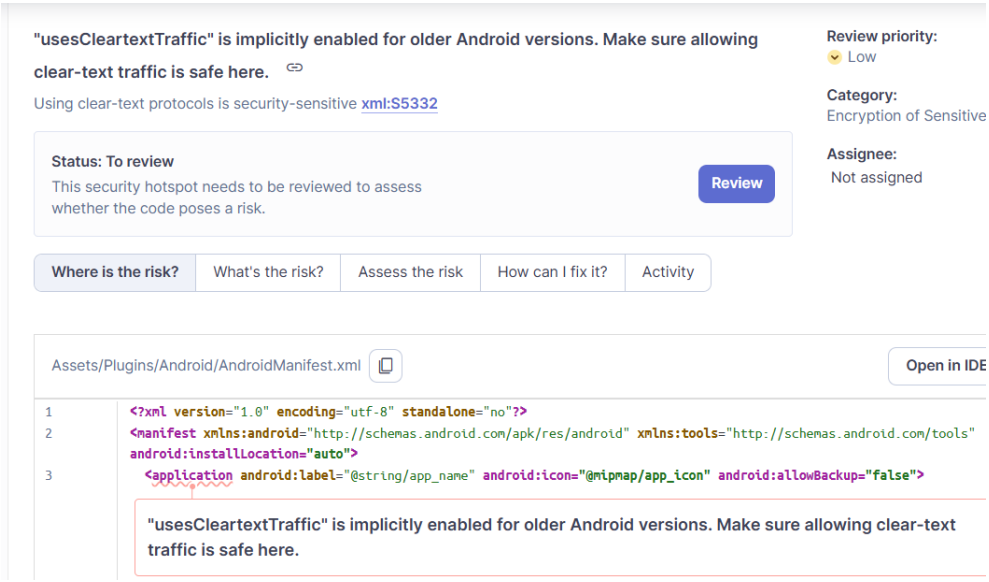


Figura 28: Revisión de Security Hotspots en SonarQube

Al analizar el código con SonarQube, se detectó un *Security Hotspot* relacionado con la propiedad usesCleartextTraffic (ver Figura 28), que está implícitamente habilitada en versiones antiguas de Android. Este ajuste puede permitir tráfico en texto claro, lo que representa un riesgo de seguridad si se transmiten datos sensibles sin cifrar. SonarQube clasifica esta alerta como de baja prioridad y recomienda revisar si el uso de tráfico sin cifrar es seguro en el contexto del proyecto. Para resolver esta advertencia, se sugiere deshabilitar explícitamente el tráfico en texto claro en el archivo AndroidManifest.xml usando android:usesCleartextTraffic="false" dentro de la etiqueta <application>, a menos que haya una justificación clara y segura para permitirlo.

Todos los errores detectados por SonarQube están relacionados únicamente con configuraciones globales del sistema, específicamente del archivo de manifiesto de Android, y no con la lógica o implementación del código desarrollada por el equipo. Esto significa que no son fallos en la programación realizada, sino ajustes predeterminados del entorno que requieren revisión para asegurar el cumplimiento de buenas prácticas de seguridad.

3.4.1 Métrica Halstead Complexity (complejidad de desarrollo)

La métrica Halstead Complexity son un conjunto de medidas propuestas por Maurice Halstead para evaluar la complejidad de un programa. Estas métricas se basan en el número de operadores y operandos distintos del código, y se utilizan para estimar el esfuerzo necesario para desarrollarlo y mantenerlo. Se fundamentan en el conteo de operadores (símbolos y palabras clave que realizan acciones) y operandos (variables y constantes sobre las que se actúa). A partir del número de operadores y operandos únicos y totales, se derivan diversas métricas como la longitud del programa, el tamaño del vocabulario, el volumen (cantidad de información contenida en el código), la dificultad (relacionada con la propensión a errores), el nivel del programa (inversamente proporcional a la dificultad), el esfuerzo de implementación, el tiempo estimado de desarrollo y una predicción del número de errores que podrían presentarse en la entrega [82].

Estas métricas buscan proporcionar una evaluación objetiva de la complejidad del código, ayudando a identificar áreas potencialmente difíciles de mantener, probar y con mayor probabilidad de contener errores. Además, permiten monitorear tendencias de complejidad a lo largo del tiempo, siendo útiles como herramientas de mantenimiento para analizar el código existente.

*a. **Understand***

Understand es una herramienta que incorpora diversas métricas para el análisis de software. Estas métricas han evolucionado con el tiempo en función de las necesidades y solicitudes de sus usuarios. Una de sus principales fortalezas radica en la flexibilidad que ofrece, ya que dispone de una API completa para su base de datos y un sistema de plugins completamente desarrollado. Esto permite a los usuarios crear métricas personalizadas en caso de que las métricas predefinidas no se ajusten a sus requerimientos específicos. La capacidad de extender y personalizar métricas a través de sus interfaces de programación la convierte en una opción robusta para proyectos que demandan análisis a medida [83].

Tabla 17: Resumen de Understand: funciones, ventajas y enfoque técnico

Aspecto	Resumen
Características	Análisis estático, visualización de arquitectura, métricas y dependencias
Ventajas	Análisis estático, visualización de arquitectura, métricas y dependencias
Desventajas	Costo elevado, interfaz antigua, curva de aprendizaje
Casos ideales	Ingeniería inversa, mantenimiento de código legacy, refactorización
Requisitos técnicos	Motor propio, base optimizada, extensible en Python y C++

La Tabla 17**,** resume las funcionalidades esenciales, ventajas, limitaciones y aspectos técnicos de Understand, una herramienta especializada en análisis estático de código. Está diseñada para facilitar tareas complejas como la ingeniería inversa, el mantenimiento de sistemas legacy y la documentación automática en proyectos de gran escala, gracias a su motor de análisis propio y su capacidad de extensión avanzada.

b. ***Cppcheck***

Cppcheck es una herramienta de análisis estático diseñada para código en C, C++ y C#. Su objetivo principal es realizar un análisis profundo para detectar errores, con un enfoque particular en la identificación de comportamientos indefinidos y construcciones peligrosas dentro del código. Una de sus características clave es la capacidad de generar muy pocos falsos positivos, lo que mejora la precisión del análisis. Cppcheck está preparado para analizar código incluso cuando se utilizan sintaxis no estándar del lenguaje. Además, proporciona tanto una interfaz de línea de comandos como una interfaz gráfica, lo que facilita su uso en distintos entornos de desarrollo. Su enfoque principal está orientado a la detección de comportamientos que no están definidos por el estándar de C/C++/C# [84].

Tabla 18: Resumen de Cppcheck: análisis, beneficios y aplicación técnica

Aspecto	**Resumen**
Características	Análisis estático de C/C++, detección de errores, CLI y GUI
Ventajas	Código abierto, ligero, fácil integración, alta precisión
Desventajas	Funcionalidad limitada, interfaz básica, sin métricas de calidad
Casos ideales	Proyectos C/C++ ligeros, CI/CD, sistemas embebidos, validación rápida
Requisitos técnicos	Análisis sin compilación, configurable, soporte XML, multihilo

Se observa en Tabla 18, las principales características, ventajas, limitaciones y requisitos técnicos de Cppcheck, una herramienta de análisis estático centrada en lenguajes como C y C++. Su enfoque en precisión y bajo consumo de recursos la convierte en una opción ideal para proyectos con restricciones computacionales o que requieren una validación rápida del código, especialmente en entornos de integración continua.

c. Mapa comparativo

Tabla 19: Mapa Comparativo Understand vs Cppcheck

Aspecto	Understand	Cppcheck
Tipo	Análisis estático completo	Análisis estático específico
Enfoque principal	Comprensión de código	Detección de errores
Lenguajes	Multilenguaje (C++, Java, C#	C, C++, C#
Métricas	Amplia variedad	Limitadas
Precio	Depende de la versión	Gratuito
Licencia	Propietaria	GPL (código abierto)
Interfaz	GUI avanzada	CLI + GUI básica
Visualización	Diagramas arquitectura complejos	Reportes básicos
Detección bugs	Moderada	Alta
Falsos positivos	Moderados	Bajos
Velocidad	Moderada	Rápida

- **Selección de herramienta y aplicación sobre el código del proyecto**

Se optó por utilizar la herramienta Understand debido a su capacidad avanzada para realizar análisis estático de código fuente con un alto nivel de detalle y precisión (ver Tabla 19). Understand ofrece un soporte robusto para la generación automática de métricas como Halstead, McCabe y otras relacionadas con la calidad y mantenibilidad del software, lo cual facilita su integración en procesos de evaluación académica e investigativa. Aunque existen herramientas alternativas como Cppcheck, esta se enfoca principalmente en la detección de errores y advertencias en código C/C++, y no proporciona de forma nativa el conjunto completo de métricas Halstead. Por tanto, se eligió Understand por ser una solución más completa y especializada para el análisis cuantitativo de la complejidad del software.

Tabla 20: Métrica Halstead Complexity del script AsignarInstrumento.cs

Script	Halstead Complexity	Métrica
AsignarInstrumento.cs	Halstead Difficulty(D)	30.18
	Halstead Effort (E)	86,720.41
	Halstead Volume (V)	2,873.27

El script AsignarInstrumento.cs presentó una dificultad Halstead (D) de 30.18, lo cual indica una lógica moderadamente compleja. El esfuerzo (E) requerido fue de 86,720.41, lo que refleja un nivel significativo de esfuerzo cognitivo para comprender y modificar el código. El volumen (V) fue de 2,873.27, evidenciando un tamaño considerable del algoritmo implementado (ver Tabla 20).

Tabla 21: Métrica Halstead Complexity del script AsignarInstrumentoGeneral.cs

Script	**Halstead Complexity**	**Métrica**
AsignarInstrumentoGeneral.cs	Halstead Difficulty(D)	19.25
	Halstead Effort (E)	17,140.45
	Halstead Volume (V)	890.41

El script AsignarInstrumentoGeneral.cs tuvo una dificultad Halstead de 19.25, lo que implica menor complejidad en comparación con otros scripts. El esfuerzo Halstead fue de 17,140.45, y el volumen fue de 890.41, mostrando que es un script más compacto y accesible para mantenimiento (ver Tabla 21).

Tabla 22: Métrica Halstead Complexity del script BotonExpandir.cs

Script	**Halstead Complexity**	**Métrica**
BotonExpandir.cs	Halstead Difficulty(D)	7.50
	Halstead Effort (E)	1,341.10
	Halstead Volume (V)	178.81

Este script mostró una dificultad de 7.50, siendo el más bajo de los analizados, lo que indica una baja complejidad. El esfuerzo necesario fue de 1,341.10, y su volumen fue de solo 178.81, lo que lo hace altamente legible y sencillo de modificar (ver Tabla 22).

Tabla 23: Métrica Halstead Complexity del script BotonInstrumento.cs

Script	**Halstead Complexity**	**Métrica**
BotonInstrumento.cs	Halstead Difficulty(D)	10.66
	Halstead Effort (E)	3,903.96
	Halstead Volume (V)	366.30

El script BotonInstrumento.cs registró una dificultad de 10.66 y un esfuerzo de 3,903.96, con un volumen de 366.30. Estas métricas reflejan una lógica simple y una implementación directa, sin demasiada carga cognitiva (ver Tabla 23).

Tabla 24: Métrica Halstead Complexity del script BotonInstrumentoGeneral.cs

Script	Halstead Complexity	Métrica
BotonInstrumentoGeneral.cs	Halstead Difficulty(D)	23.54
	Halstead Effort (E)	25,468.22
	Halstead Volume (V)	1,081.89

Con una dificultad de 23.54 y un esfuerzo de 25,468.22, este script muestra un nivel de complejidad y trabajo relativamente altos. El volumen de 1,081.89 respalda esta evaluación, sugiriendo un código de mayor extensión y procesamiento (ver Tabla 24).

Tabla 25: Métrica Halstead Complexity del script CambiarMenus.cs

Script	Halstead Complexity	Métrica
CambiarMenus.cs	Halstead Difficulty(D)	26.43
	Halstead Effort (E)	44,061.05
	Halstead Volume (V)	1,667.17

Este script tuvo una dificultad de 26.43, un valor relativamente alto. El esfuerzo calculado fue de 44,061.05, y su volumen fue de 1,667.17, lo cual indica que involucra una lógica interna más elaborada y requiere mayor concentración para su entendimiento o modificación (ver Tabla 25).

Tabla 26: Métrica Halstead Complexity del script CanvasController.cs

Script	Halstead Complexity	Métrica
CanvasController.cs	Halstead Difficulty(D)	16.04
	Halstead Effort (E)	7,183.98
	Halstead Volume (V)	447.83

El script CanvasController.cs presentó una dificultad de 16.04 y un esfuerzo de 7,183.98. Con un volumen de 447.83, se considera de complejidad moderada y accesible para desarrolladores con experiencia en la base de código (ver Tabla 26).

Tabla 27: Métrica Halstead Complexity del script CargaAsincrona.cs

Script	Halstead Complexity	Métrica
CargaAsincrona.cs	Halstead Difficulty(D)	16.41
	Halstead Effort (E)	14,673.42
	Halstead Volume (V)	894.16

CargaAsincrona.cs tuvo una dificultad de 16.41, un esfuerzo de 14,673.42 y un volumen de 894.16. Estas métricas indican una implementación razonablemente estructurada con complejidad media, adecuada para tareas de carga o gestión asincrónica de datos (ver Tabla 27).

Tabla 28: Métrica Halstead Complexity del script ControladorExpandir.cs

Script	Halstead Complexity	Métrica
ControladorExpandir.cs	Halstead Difficulty(D)	15.00
	Halstead Effort (E)	9,638.20
	Halstead Volume (V)	642.55

Este script presentó una dificultad Halstead de 15.00, con un esfuerzo de 9,638.20 y un volumen de 642.55. Estos valores reflejan una lógica de complejidad media, manejable en términos de lectura y mantenimiento del código (ver Tabla 28).

Tabla 29: Métrica Halstead Complexity del script ControlMesaMayo.cs

Script	Halstead Complexity	Métrica
ControlMesaMayo.cs	Halstead Difficulty(D)	28.38
	Halstead Effort (E)	44,718.56
	Halstead Volume (V)	1,575.80

Con una dificultad de 28.38, este script demuestra una complejidad considerable. El esfuerzo necesario fue de 44,716.56 y el volumen alcanzó los 1,575.80, lo cual sugiere una estructura elaborada que podría requerir mayor tiempo de comprensión por parte de un desarrollador (ver Tabla 29).

Tabla 30: Métrica Halstead Complexity del script ControlTiemposGeneral.cs

Script	Halstead Complexity	Métrica
ControlTiemposGeneral.cs	Halstead Difficulty(D)	58.29
	Halstead Effort (E)	397,330.29
	Halstead Volume (V)	6,817.01

El script más complejo del conjunto, ControlTiemposGeneral.cs, tuvo una dificultad de 58.29 y un esfuerzo muy alto de 397,330.29. Su volumen fue de 6,817.01, lo que indica que contiene una lógica extensa y detallada, representando un punto crítico en cuanto a mantenibilidad y complejidad cognitiva (ver Tabla 30).

Tabla 31: Métrica Halstead Complexity del script LoadSceneWithMessage.cs

Script	Halstead Complexity	Métrica
LoadSceneWithMessage.cs	Halstead Difficulty(D)	21.67
	Halstead Effort (E)	22,554.05
	Halstead Volume (V)	1,040.96

Con una dificultad de 21.67, un esfuerzo de 22,554.05 y un volumen de 1,040.96, este script refleja una complejidad media que puede ser manejada con relativa facilidad, aunque con cierta carga cognitiva durante la revisión o modificación (ver Tabla 31).

Tabla 32: Métrica Halstead Complexity del script MenuPrincipalUta.cs

Script	Halstead Complexity	Métrica
MenuPrincipalUta.cs	Halstead Difficulty(D)	12.13
	Halstead Effort (E)	14,591.77
	Halstead Volume (V)	1,202.72

Este script tuvo una dificultad de 12.13, un esfuerzo de 14,591.77 y un volumen de 1,202.72. Los valores indican una estructura de complejidad baja a media, enfocada probablemente en tareas de navegación o interfaz (ver Tabla 32).

Tabla 33: Métrica Halstead Complexity del script MostrarNombreObjetoVR.cs

Script	Halstead Complexity	Métrica
MostrarNombreObjetoVR.cs	Halstead Difficulty(D)	16.07
	Halstead Effort (E)	6,669.64
	Halstead Volume (V)	415.00

Con una dificultad de 16.07, un esfuerzo de 6,669.64 y un volumen de 415.00, este script muestra una lógica contenida y bien delimitada, de baja complejidad, ideal para funciones auxiliares o complementarias dentro de la aplicación (ver Tabla 33).

Tabla 34: Métrica Halstead Complexity del script MultiplayerCreationControllerGeneral.cs

Script	Halstead Complexity	Métrica
MultiplayerCreationControllerGeneral.cs	Halstead Difficulty(D)	32.10
	Halstead Effort (E)	108,462.84
	Halstead Volume (V)	3,379.38

Este script presentó una dificultad de 32.10, una de las más altas del análisis, con un esfuerzo de 108,462.84 y un volumen de 3,379.38. Los valores reflejan una funcionalidad extensa y compleja, posiblemente relacionada con la gestión de sesiones multijugador, que requiere atención especial en su mantenimiento (ver Tabla 34).

Tabla 35: Métrica Halstead Complexity del script ShowObjectNameOnHover.cs

Script	Halstead Complexity	Métrica
ShowObjectNameOnHover.cs	Halstead Difficulty(D)	17.69
	Halstead Effort (E)	13,559.71
	Halstead Volume (V)	766.53

El script tuvo una dificultad de 17.69, un esfuerzo de 13,559.71 y un volumen de 766.53, lo que lo posiciona como una funcionalidad de mediana complejidad, bien contenida y específica, centrada en interacción con objetos (ver Tabla 35).

Tabla 36: Métrica Halstead Complexity del script UIInteractionController.cs

Script	Halstead Complexity	Métrica
UIInteractionController.cs	Halstead Difficulty(D)	29.46
	Halstead Effort (E)	35,586.53
	Halstead Volume (V)	1,207.79

Finalmente, este script presentó una dificultad de 29.46, un esfuerzo de 35,586.53 y un volumen de 1,207.79. Estas métricas lo sitúan como un componente importante en la lógica de interacción de la interfaz, con un nivel de complejidad significativo que debe ser gestionado cuidadosamente en el proceso de desarrollo (ver Tabla 36).

3.5 FPS Frames Per Second (Rendimiento en Tiempo Real)

La medición de fotogramas por segundo (FPS) constituye una métrica fundamental para evaluar el rendimiento de aplicaciones de realidad virtual, ya que determina directamente la fluidez de la experiencia del usuario y su sensación de presencia en el entorno virtual [85], [86].

Los fotogramas por segundo (FPS) miden la cantidad de imágenes fijas, o fotogramas, que se muestran en un segundo de vídeo o animación. Cuanto mayor sea el número de fotogramas, más fluido y realista será el movimiento para el espectador. En general, una velocidad de fotogramas de 30 FPS o superior se considera fluida, mientras que una velocidad inferior a 30 FPS puede parecer entrecortada o irregular [86].

Herramientas como Unity Profiler, Unreal Insights y OVR Metric Tools permiten analizar en detalle los FPS (Frames Per Second) y otros indicadores clave de rendimiento. Esto facilita la identificación de cuellos de botella y áreas de optimización, lo que a su vez ayuda a mejorar la fluidez y la experiencia general del usuario en tiempo real.

*a. **Unity Profiler***

Los fotogramas por segundo (FPS) miden la cantidad de imágenes fijas, o fotogramas, que se muestran en un segundo de vídeo o animación. Cuanto mayor sea el número de fotogramas, más fluido y realista será el movimiento para el espectador. En general, una velocidad de fotogramas de 30 FPS o superior se considera fluida, mientras que una velocidad inferior a 30 FPS puede parecer entrecortada o irregular [87].

Herramientas como Unity Profiler, Unreal Insights y OVR Metric Tools permiten analizar en detalle los FPS (Frames Per Second) y otros indicadores clave de rendimiento. Esto facilita la identificación de cuellos de botella y áreas de optimización, lo que a su vez ayuda a mejorar la fluidez y la experiencia general del usuario en tiempo real [88].

Tabla 37: Resumen de Unity Profiler: monitoreo de rendimiento en entornos Unity

Aspecto	Resumen
Características	Métricas en tiempo real, integración con Unity, análisis por frame/hilo
Ventajas	Integración nativa, análisis profundo, útil para VR y móviles
Desventajas	Puede afectar el rendimiento, interpretación compleja, configuración extra
Casos ideales	Optimización de juegos, VR/AR, detección de cuellos de botella
Compatibilidad	Optimización de juegos, VR/AR, detección de cuellos de botella
Soporte	Soporte oficial de Unity, comunidad activa, buena documentación

La Tabla 37**,** resume las principales funciones, beneficios, limitaciones, casos de uso y compatibilidad del Unity Profiler, una herramienta integrada en el motor Unity para el análisis y optimización del rendimiento. Su capacidad para monitorear múltiples aspectos del sistema en tiempo real, tanto en el editor como en dispositivos físicos, lo convierte en un recurso esencial en el desarrollo de aplicaciones interactivas, especialmente en entornos móviles y de realidad virtual.

*b. **OVR Metrics Tools***

OVR Metrics Tools representa un conjunto especializado de utilidades desarrolladas por Oculus para la medición precisa del rendimiento en aplicaciones de realidad virtual, con énfasis particular en el análisis de FPS y latencia en dispositivos Quest. Esta herramienta proporciona métricas críticas como la visualización de la variación temporal de los FPS, tiempos de fotograma y predicción de poses, facilitando la detección de inconsistencias que podrían provocar cinetosis en los usuarios, adicionalmente, su panel de control en tiempo real permite monitorear el rendimiento

térmico y energético del dispositivo, factores que inciden directamente en la sostenibilidad de altas tasas de fotogramas durante sesiones prolongadas de uso [89].

Tabla 38: Resumen de OVR Metrics Tool: monitoreo de rendimiento en dispositivos Meta Quest

Aspecto	**Resumen**
Características	Métricas en tiempo real, visor overlay, rendimiento térmico y energía
Ventajas	Oficial de Meta, ligera, ideal para pruebas VR móviles
Desventajas	Solo para Meta Quest, métricas limitadas, sin análisis de código
Casos ideales	Validación de apps VR, detección de throttling, pruebas de rendimiento
Compatibilidad	Exclusivo para Meta Quest, funciona con Unity/Unreal, requiere ADB inicial
Soporte	Soporte oficial de Meta, comunidad activa, documentación técnica

Como se ve en la Tabla 38, presenta un resumen de las funciones clave, ventajas, limitaciones, casos de uso y compatibilidad de OVR Metrics Tool, una herramienta oficial de Meta diseñada específicamente para monitorear el rendimiento en dispositivos Quest. Su capacidad de mostrar métricas en tiempo real directamente en el visor permite a los desarrolladores evaluar y optimizar experiencias VR sin necesidad de conexión a una PC, convirtiéndola en una solución ligera y eficaz para entornos móviles.

c. Unreal Insights

Unreal Insights es una suite de captura y análisis de telemetría que permite capturar eventos de su proyecto a altas velocidades de datos. Unreal Insights le ayuda a identificar áreas de datos que podrían requerir optimización [90].

Unreal Insights ayuda a los desarrolladores a identificar cuellos de botella, lo cual resulta útil para optimizar el rendimiento. En general, Unreal Insights es un sistema de creación de perfiles independiente que se integra con Unreal Engine para recopilar,

analizar y visualizar los datos emitidos por el motor. Además de ofrecer una cobertura robusta de los sistemas existentes en el motor, Unreal Insights facilita la incorporación de datos de creación de perfiles. Finalmente, el sistema permite registrar datos de forma remota, minimizando el impacto de la aplicación en la ejecución del proyecto [91].

Tabla 39: Resumen de Unreal Insights: análisis avanzado de rendimiento en Unreal Engine

Aspecto	**Resumen**
Características	Análisis de trazas, latencia, CPU/GPU, hilos, integración con Unreal
Ventajas	Análisis detallado, multihilo, integrado al editor
Desventajas	Curva de aprendizaje alta, archivos grandes, documentación parcial
Casos ideales	Optimización de juegos complejos, problemas de latencia rendimiento
Compatibilidad	Unreal 4.23 en adelante, multiplataforma (PC, consolas, VR, móviles)
Soporte	Soporte de Epic Games, comunidad activa, documentación y eventos

La Tabla 39, muestra un resumen técnico de Unreal Insights, una herramienta integrada en Unreal Engine para el análisis exhaustivo de rendimiento. Gracias a su capacidad para registrar y visualizar trazas detalladas del comportamiento del motor en tiempo real, esta herramienta se vuelve esencial para desarrolladores que buscan optimizar el rendimiento en proyectos de alta complejidad, como videojuegos AAA, entornos multijugador o experiencias inmersivas en realidad virtual.

d. Mapa comparativo

Tabla 40: Mapa comparativo Unity Profile, Herramienta de métricas OVR y Unreal Insights

Características	**Unity Profiler**	**Herramienta de métricas OVR**	**Unreal Insights**
Licencia	Propietaria	Propietaria	Propietaria
Compatibilidad GPU	Multivendor	Multivendor (en Quest, con chipset integrado)	Multivendor
Sistemas operativos	Windows, macOS, Android, iOS, consolas	Solo en dispositivos Meta Quest (basado en Android)	Windows, macOS, consolas, Android, iOS
APIs compatibles	Soporta múltiples APIs a través de Unity Engine	Interno, no depende directamente de APIs gráficas	APIs utilizadas por Unreal (DirectX, Vulkan, etc.)
Análisis en tiempo real	Sí	Sí (superposición en visor)	Sí (con posibilidad de grabar sesiones)
Depuración frame-by-frame	Parcial (con Frame Debugger)	No	Sí
Integración IDE	Integrado en Unity	Independiente, uso en visor	Integrado en Unreal Engine
Uso principal	Análisis de rendimiento general y por subsistema	Medición de rendimiento VR en Quest	Trazado detallado de eventos y análisis profundo

- **Selección de herramienta y aplicación sobre el código del proyecto**

Para la evaluación del rendimiento de la aplicación de realidad virtual, se seleccionó la herramienta OVR Metrics Tool debido a que como se observa en la Tabla 40 esta herramienta puede proporcionar métricas precisas y en tiempo real directamente en el visor Meta Quest. A diferencia de otras herramientas como el Unity Profiler, que

pueden introducir overhead o requerir conexión a una computadora, OVR Metrics Tool permite monitorear el desempeño de la aplicación en su entorno nativo y en condiciones reales de uso. Esto resulta especialmente útil para medir la fluidez (FPS), el uso de CPU y GPU, así como posibles limitaciones térmicas (thermal throttling), factores críticos en aplicaciones inmersivas donde la experiencia del usuario depende en gran medida del rendimiento sostenido del sistema.

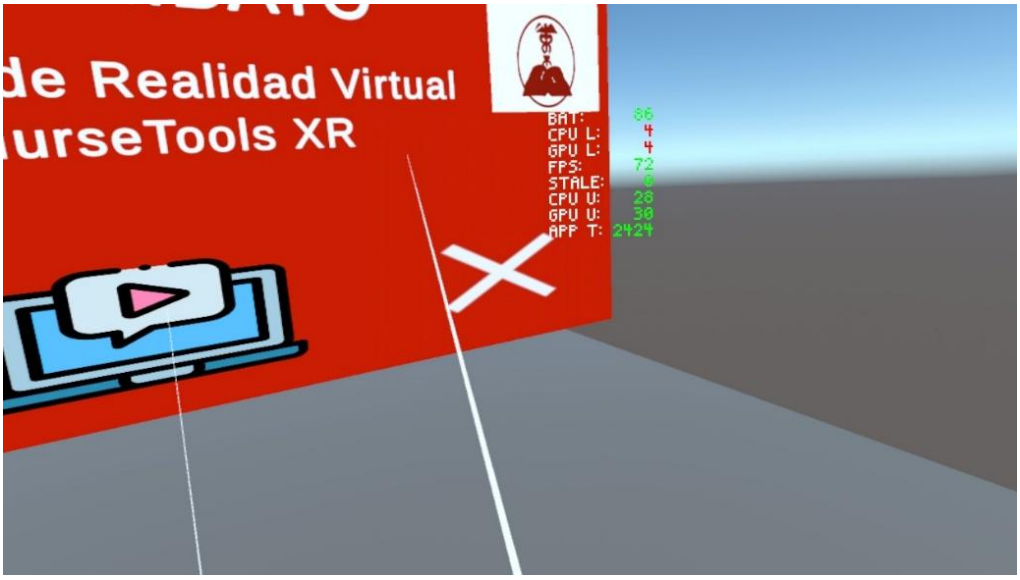


Figura 29: FPS en Menú Inicial

Durante la evaluación del menú inicial de la aplicación, se registró una tasa de FPS constante de 72 cuadros por segundo (ver Figura 29), que corresponde al límite máximo establecido por los dispositivos Meta Quest cuando la aplicación no utiliza el modo de 90Hz. Esta tasa de cuadros por segundo es un indicador positivo, ya que implica que la escena se ejecuta de manera fluida, sin caídas perceptibles que puedan afectar la experiencia del usuario. Además, el valor de “STALE” en 0 indica que no hubo cuadros omitidos, lo cual confirma la estabilidad del renderizado. Este resultado es especialmente importante en aplicaciones de realidad virtual, donde una baja tasa de FPS puede generar incomodidad o mareo en los usuarios. Por lo tanto, se concluye que el menú inicial presenta un rendimiento óptimo y adecuado para la experiencia inmersiva deseada.



Figura 30: FPS en observando todos los objetos del quirófano

Durante la evaluación del escenario quirófano virtual en su máxima complejidad, con la totalidad de los objetos médicos se observa una media de 43 FPS (ver Figura 30). La densidad de elementos tridimensionales tiene un impacto directo en el rendimiento, evidenciado por una disminución significativa en los cuadros por segundo (43 FPS), lo cual representa una caída notable respecto a escenas menos cargadas. Esta medición revela la necesidad de implementar técnicas de optimización como LOD (Level of Detail), culling o uso de texturas menos pesadas, especialmente si se desea mantener la fluidez requerida por entornos de realidad virtual inmersiva sin comprometer el nivel de detalle visual necesario para simulaciones educativas realistas.

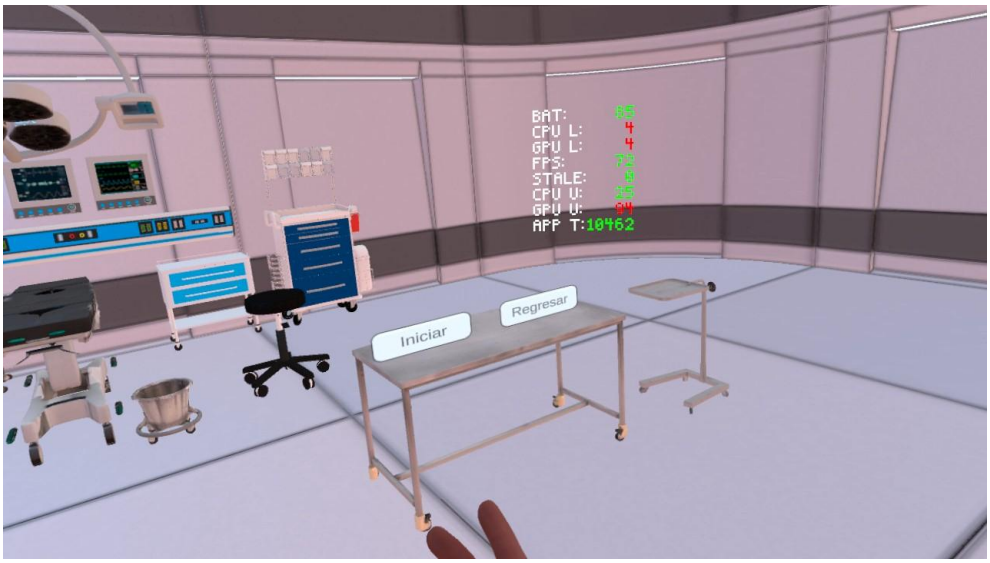


Figura 31: FPS sin muchos objetos

Durante la evaluación del quirófano virtual sin visualizar todos los objetos de la escena al mismo tiempo se observa una media de 72 FPS (ver Figura 31). A pesar de la densidad de objetos, el rendimiento del sistema es notablemente eficiente, con una tasa de refresco de 72 FPS, lo que indica una experiencia fluida para el usuario. Este valor es ideal para aplicaciones de VR, ya que reduce el riesgo de mareo y mejora la inmersión. El bajo uso del CPU y GPU en estado "L" (latente) refleja una buena optimización del entorno, mientras que los valores activos (CPU V: 25 y GPU V: 94) sugieren que, aunque hay una carga considerable, el sistema puede manejarla adecuadamente. Esto valida la viabilidad del entorno quirúrgico con alta cantidad de objetos sin comprometer la experiencia del usuario.

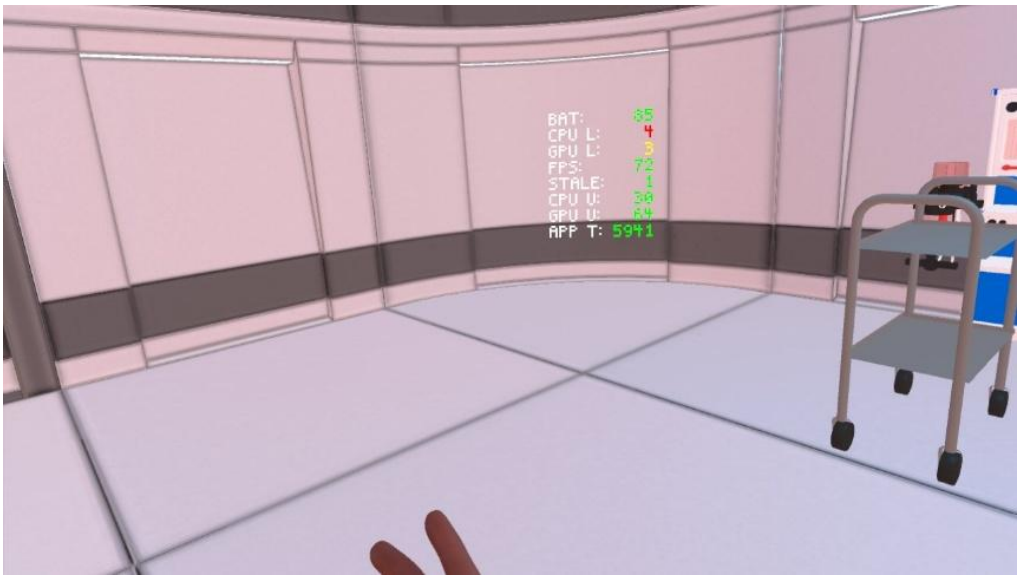


Figura 32: FPS con muy pocos objetos

En esta situación la escena presenta un entorno más despejado, con apenas algunos elementos visibles y mayor espacio libre. Aun en esta configuración más ligera, la tasa de FPS se mantiene constante en 72 FPS (ver Figura 32), lo cual demuestra la estabilidad del rendimiento del sistema independientemente de la cantidad de objetos renderizados. El uso del GPU en estado activo disminuye significativamente (GPU V: 50), lo que indica una menor carga gráfica y, por ende, una eficiencia energética superior. Estos resultados permiten concluir que el sistema está bien optimizado para distintos niveles de complejidad visual, garantizando una experiencia VR estable tanto en escenarios simples como complejos, aspecto fundamental en aplicaciones educativas inmersivas como la simulación de procedimientos médicos.

3.5.1 Uso de CPU/GPU (Rendimiento en Tiempo Real)

El uso de CPU y GPU en aplicaciones es fundamental para garantizar un rendimiento eficiente, especialmente en tareas que requieren procesamiento intensivo como los gráficos en tiempo real, inteligencia artificial y simulaciones complejas. La CPU (Unidad Central de Procesamiento) se encarga de operaciones generales y de control del sistema, mientras que la GPU (Unidad de Procesamiento Gráfico) está optimizada para ejecutar cálculos paralelos masivos, lo que la hace ideal para renderizado de gráficos y aprendizaje profundo[92].

Herramientas como NVIDIA Nsight y RenderDoc permiten analizar el uso de estos recursos en detalle, ayudando a optimizar el rendimiento del sistema en tiempo real [93], [94].

a. NVIDIA Nsight

NVIDIA Nsight es una herramienta desarrollada por la empresa NVIDIA, destinada al análisis, depuración y optimización de aplicaciones que hacen uso intensivo de la GPU. Permite a los desarrolladores visualizar en detalle la interacción entre la CPU y la GPU durante la ejecución de una aplicación, lo que facilita la identificación de cuellos de botella, fallos gráficos y problemas de rendimiento. Nsight ofrece compatibilidad con diversas APIs como Vulkan, OpenGL, Direct3D, OpenVR y Oculus SDK, siendo útil en una amplia gama de proyectos que requieren procesamiento gráfico en tiempo real, como videojuegos, simulaciones o entornos de realidad virtual [94].

Además, esta herramienta proporciona la posibilidad de capturar cuadros individuales (frames), examinar los estados del pipeline gráfico. Su integración con entornos de desarrollo como Visual Studio permite a los desarrolladores realizar análisis detallados sin salir del flujo de trabajo principal. Gracias a su enfoque profesional y completo, NVIDIA Nsight se ha consolidado como una solución clave para quienes necesitan mejorar la eficiencia y estabilidad de sus aplicaciones gráficas [94].

Tabla 41: Resumen de NVIDIA Nsight: análisis gráfico avanzado para entornos de alto rendimiento

Aspecto	**Resumen**
Características	Monitoreo CPU/GPU, análisis de shaders, inspección de frames, soporte multigráfica
Ventajas	Análisis profundo, optimización para NVIDIA
Desventajas	Solo para GPUs NVIDIA, curva de aprendizaje elevada
Casos ideales	Videojuegos AAA, simulaciones, realidad virtual, IA acelerada por GPU
Compatibilidad	Windows/Linux, APIs gráficas (Vulkan, Direct3D, OpenGL, CUDA)
Soporte	Soporte oficial de NVIDIA, documentación extensa, comunidad activa

Como se observa en Tabla 41, las capacidades principales de NVIDIA Nsight, una herramienta especializada en el análisis gráfico avanzado y la optimización de aplicaciones con alto requerimiento computacional. Su integración con múltiples APIs y compatibilidad con plataformas de desarrollo profesionales la posicionan como una solución robusta y precisa para entornos donde se busca explotar al máximo el rendimiento del hardware NVIDIA, especialmente en el desarrollo de videojuegos, realidad virtual e inteligencia artificial.

b. RenderDoc

RenderDoc es una herramienta de código abierto diseñada para la captura y depuración de cuadros en aplicaciones gráficas que utilizan APIs como Vulka. Es una herramienta preferida para diagnosticar problemas de renderizado, la identificación de artefactos visuales, problemas de rendimiento y comportamientos inesperados en la gráfica, ofreciendo una visión transparente del funcionamiento interno del motor de renderizado sin necesidad de modificar el código fuente de la aplicación en todas nuestras aplicaciones y API compatibles [93].

Gracias a su interfaz intuitiva y su potente funcionalidad, RenderDoc se ha convertido en una herramienta ampliamente adoptada tanto por profesionales como por desarrolladores independientes. A pesar de ser gratuita, ofrece características de nivel

avanzado que permiten un diagnóstico gráfico profundo. Su continuo desarrollo por parte de la comunidad y su compatibilidad con motores de juego y plataformas modernas la posicionan como una solución confiable y eficiente para el análisis de gráficos en tiempo real [93].

Tabla 42: Características y usos de RenderDoc: herramienta de depuración gráfica multiplataforma

Aspecto	**Resumen**
Características	Captura de frames, análisis de shaders y pipeline, código abierto, interfaz intuitiva
Ventajas	Gratuita, fácil de usar, soporte multiplataforma
Desventajas	No enfocada en análisis de CPU o cómputo general por GPU
Casos ideales	Depuración y optimización gráfica, análisis educativo
Compatibilidad	Windows y Linux, soporte para Vulkan, DirectX, OpenGL, y otros
Soporte	Comunidad activa, documentación abierta, integración con Unity y Unreal

En la Tabla 42**,** resume las funcionalidades de RenderDoc, una herramienta de análisis gráfico ampliamente utilizada por desarrolladores para depurar y optimizar aplicaciones visuales. Su naturaleza de código abierto, junto con su facilidad de uso y soporte para múltiples APIs gráficas, la convierten en una solución versátil tanto para entornos profesionales como educativos, especialmente en tareas de análisis de shaders, texturas y pipelines de renderizado.

c. Mapa comparativo

En la Tabla 43, se presenta una comparación entre las herramientas de análisis gráfico **NVIDIA Nsight** y **RenderDoc**, destacando sus principales diferencias en aspectos como el desarrollador, tipo de licencia, compatibilidad con GPUs, sistemas operativos soportados, APIs compatibles y funcionalidades específicas. Mientras que NVIDIA Nsight ofrece una integración profunda con Visual Studio, análisis en tiempo real y depuración frame-by-frame, RenderDoc se caracteriza por ser una solución de código abierto, independiente, con una interfaz más intuitiva. Ambas herramientas están orientadas a la optimización de GPU y el análisis profundo del rendimiento gráfico.

Tabla 43: Mapa comparativo NVIDIA Nsight vs RenderDoc

Característica	NVIDIA Nsight	RenderDoc
Desarrollador	NVIDIA	Balazs Tóth / Khronos Group
Licencia	Propietaria	Código abierto (MIT)
Compatibilidad GPU	NVIDIA	Multivendor (NVIDIA, AMD, Intel)
Sistemas operativos	Windows, Linux	Windows, Linux
APIs compatibles	DirectX, Vulkan	Vulkan, DirectX, OpenGL
Análisis en tiempo real	Sí	No
Depuración frame-by-frame	Sí	Sí
Integración IDE	Visual Studio	Independiente
Interfaz	Compleja, profesional	Intuitiva, ligera
Uso principal	Optimización GPU, análisis profundo	Optimización GPU, análisis profundo

- **Selección de herramienta y aplicación sobre el código del proyecto**

Después de toda la investigación acerca de las métricas de tiempo real (ver Tabla 43), se escogió NVIDIA Nsight como la herramienta más adecuada para el proyecto. Esta decisión se basó en su capacidad para ofrecer análisis detallados del rendimiento en aplicaciones gráficas y en tiempo real. Nsight permite identificar cuellos de botella, optimizar recursos y mejorar la eficiencia del sistema. Además, su compatibilidad con entornos de desarrollo como Unity y motores gráficos avanzados la convierte en una opción ideal. En consecuencia, se consideró la herramienta más eficiente para monitorear y optimizar el rendimiento en escenarios simulados.

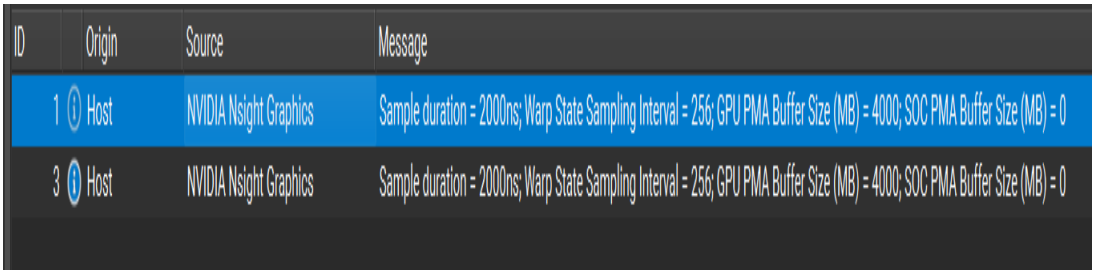


Figura 33: Resultados de Monitoreo de Rendimiento en Tiempo Real con NVIDIA Nsight Graphics

En la Figura 33, muestra la ventana de mensajes de salida generada por NVIDIA Nsight Graphics tras el análisis del sistema en tiempo real. Se observan parámetros clave como la duración de muestreo (2000ns), el intervalo de muestreo del estado de warp (256) y el tamaño del búfer de la GPU PMA (4000 MB), mientras que el búfer del SOC PMA se mantiene en 0 MB. Estos datos permiten evaluar el uso de recursos gráficos y optimizar el rendimiento del entorno de simulación.

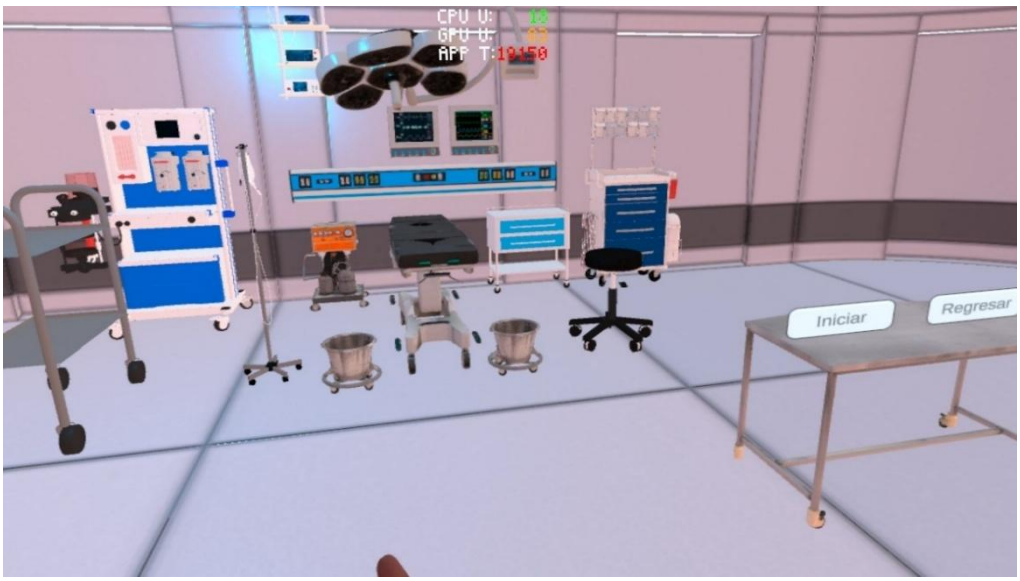


Figura 34:Monitoreo de Rendimiento en Simulación de Quirófano Virtual

La Figura 34, muestra una simulación del quirófano en un entorno virtual, pero el enfoque principal está en los valores de rendimiento que aparecen en la parte superior de la pantalla. Estos indicadores son esenciales para evaluar el uso de los recursos del sistema durante la ejecución de la simulación:

CPU V: 16 – Este valor indica el porcentaje de uso actual del procesador (CPU). Un 16% sugiere que la carga del procesador es baja, lo que implica un buen rendimiento y fluidez en la simulación.

GPU V: 83 – Este es el porcentaje de uso de la tarjeta gráfica (GPU). Un 83% indica una alta demanda gráfica, lo cual es común en simulaciones 3D complejas como esta, ya que la GPU es responsable de renderizar todos los gráficos en tiempo real.

APP T: 19150 – Este valor representa probablemente el tiempo de ejecución acumulado de la aplicación en milisegundos (ms), es decir, 19.15 segundos desde que se inició la simulación.



Figura 35: Supervisión de Recursos en una Simulación XR Médica

Como se muestra en la Figura 35, una simulación médica en realidad extendida (XR) con la aplicación "NurseTools XR", enfocándose en el monitoreo del rendimiento del sistema. En la parte superior se observan tres indicadores clave: CPU V (55), que muestra un uso moderado del procesador; GPU V (95), que indica un uso muy alto de la tarjeta gráfica debido al procesamiento gráfico intenso típico de entornos XR; y APP T (19706), que representa el tiempo de ejecución de la aplicación en milisegundos, equivalente a 19.7 segundos. Estos valores permiten evaluar el impacto de la simulación sobre el hardware y ayudan a detectar posibles cuellos de botella en el rendimiento.

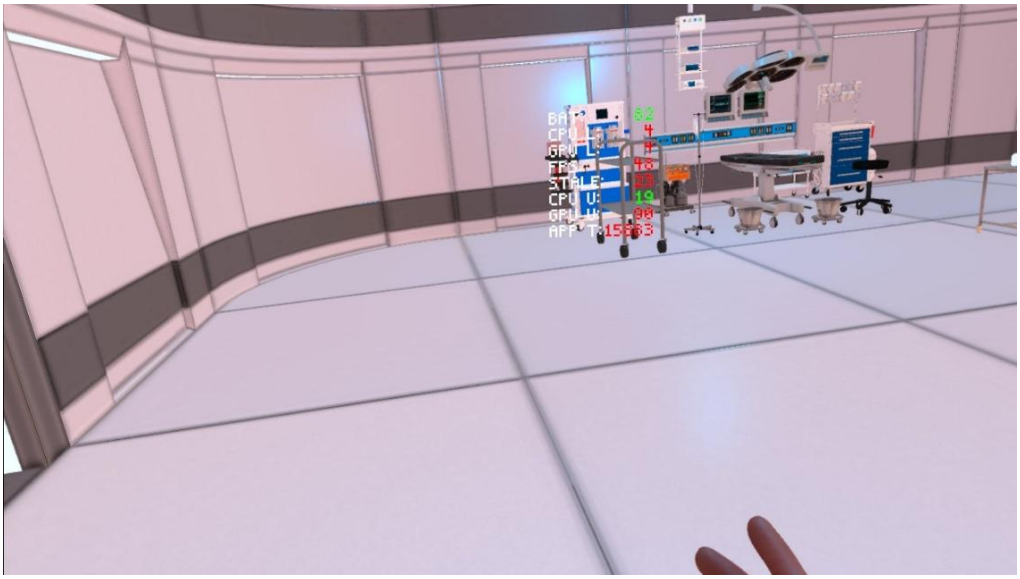


Figura 36:Análisis de Rendimiento en una Simulación Médica 3D

La imagen muestra un entorno virtual detallado que representa una sala de operaciones, compuesta por elementos como luces quirúrgicas, monitores médicos y carros de instrumental, todo renderizado en tres dimensiones (ver Figura 36). En la esquina superior derecha se presentan varios indicadores de rendimiento del sistema, los cuales permiten monitorear el estado del hardware durante la ejecución de la aplicación.

El usuario puede observar valores como BAT: 82, lo cual probablemente indique un nivel de batería del 82%, y FPS: 48, lo que significa que se están procesando 48 cuadros por segundo, ofreciendo una experiencia relativamente fluida, aunque mejorable para escenas más exigentes o aplicaciones de realidad virtual. Los valores CPU L: 4 y GPU L: 4 podrían referirse al número de núcleos lógicos activos en cada procesador, mientras que CPU U: 19 indica que la CPU está utilizando el 19% de su capacidad total, lo que refleja una carga baja. Por otro lado, GPU U: 90 revela que la GPU está trabajando al 90% de su capacidad, lo que sugiere que es el componente más solicitado en esta escena. Finalmente, STALE: 23 podría hacer referencia a cuadros o tareas desactualizadas, lo que implica cierta demora o falta de sincronización entre los procesos, posiblemente afectando la fluidez percibida.

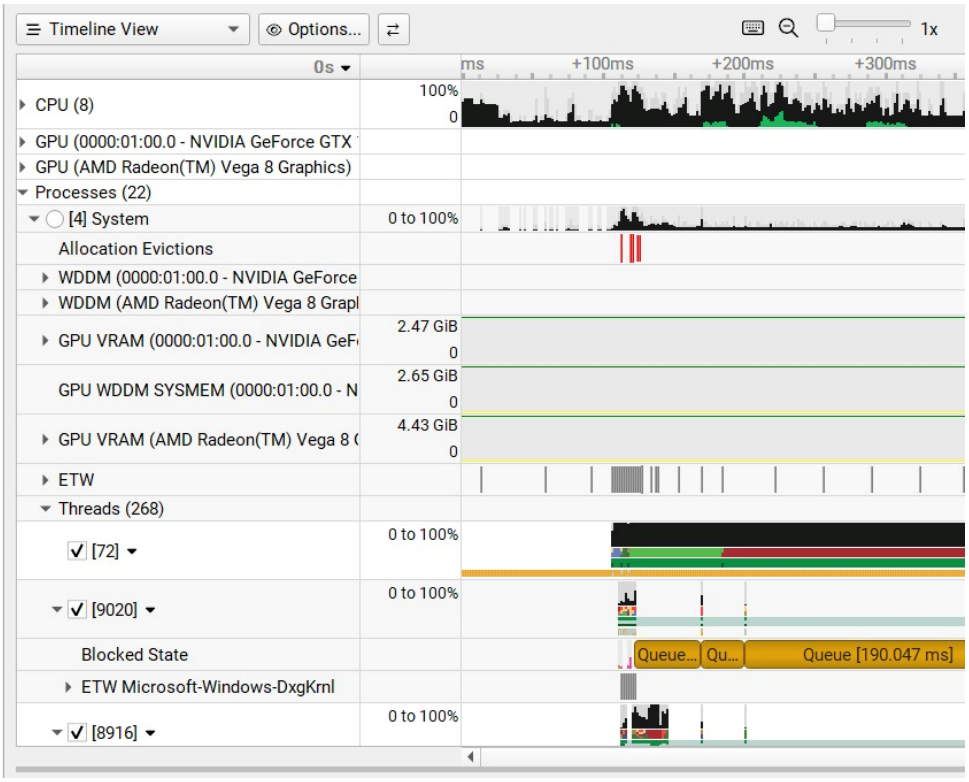


Figura 37: Carga de GPU y CPU con múltiples colas de espera

Tal como se observa en la Figura 37, una alta carga en la CPU y actividad constante en las GPUs NVIDIA GeForce y AMD Radeon. Las colas en amarillo indican cuellos de botella de hasta 190 ms en el procesamiento de tareas del sistema. El hilo identificado con PID 9020 refleja actividad intensa en la GPU, lo cual sugiere demoras relacionadas con la ejecución de tareas gráficas, se observa una vista cronológica del uso del CPU (8 núcleos) y las dos GPUs. La carga del CPU se muestra con picos de utilización hasta el 100%, mientras que las GPUs tienen actividad repartida, con la Vega 8 alcanzando hasta 4.43 GiB de uso de VRAM. En los hilos del proceso [9020], se aprecia una alta carga de trabajo encolada (con etiquetas de Queue), indicando esperas prolongadas que podrían estar afectando el rendimiento general.

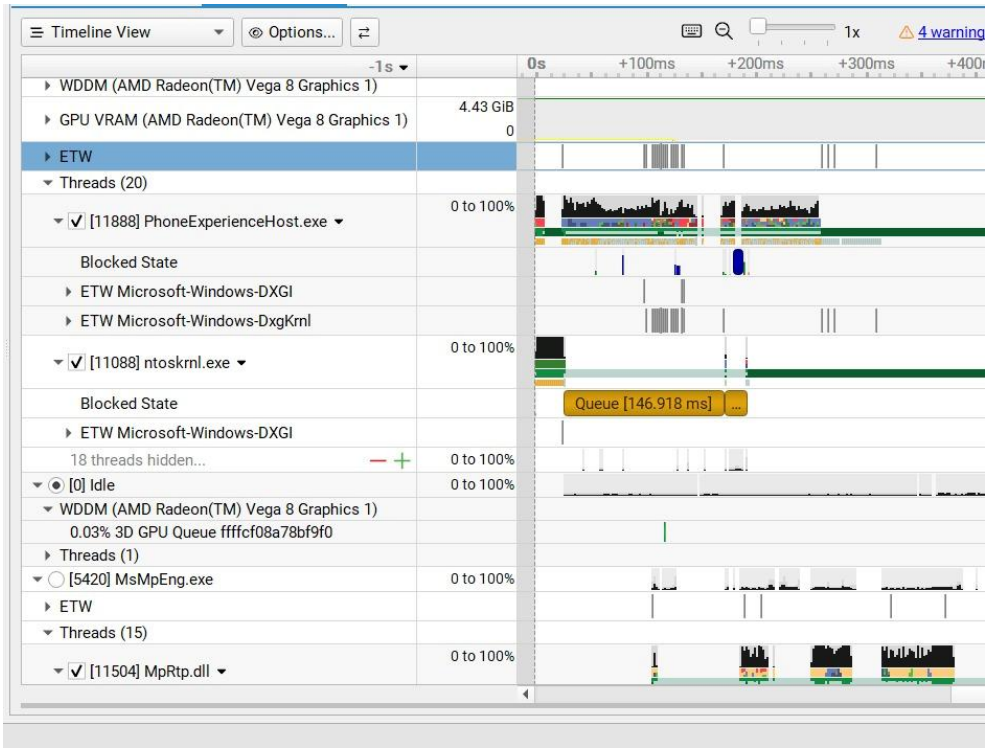


Figura 38: Bloqueo de hilos por colas de GPU en procesos del sistema

Como se muestra en la Figura 38, se destaca el proceso PhoneExperienceHost.exe y ntoskrnl.exe, con bloqueos causados por la cola de la GPU (hasta 146 ms). Este tipo de esperas indica que el sistema está atascado esperando a que la GPU libere recursos. Además, se muestra el uso completo de la memoria de video VRAM en la AMD Radeon Vega 8, se visualiza una carga sostenida sobre el proceso PhoneExperienceHost.exe y el núcleo del sistema (ntoskrnl.exe). Destaca nuevamente la presencia de colas de procesamiento con duraciones superiores a 140 ms, lo que sugiere cuellos de botella en la planificación de tareas o en la comunicación con la GPU. Además, se muestran procesos en estado Idle, lo cual indica recursos que no están siendo completamente aprovechados a pesar de las colas de espera.

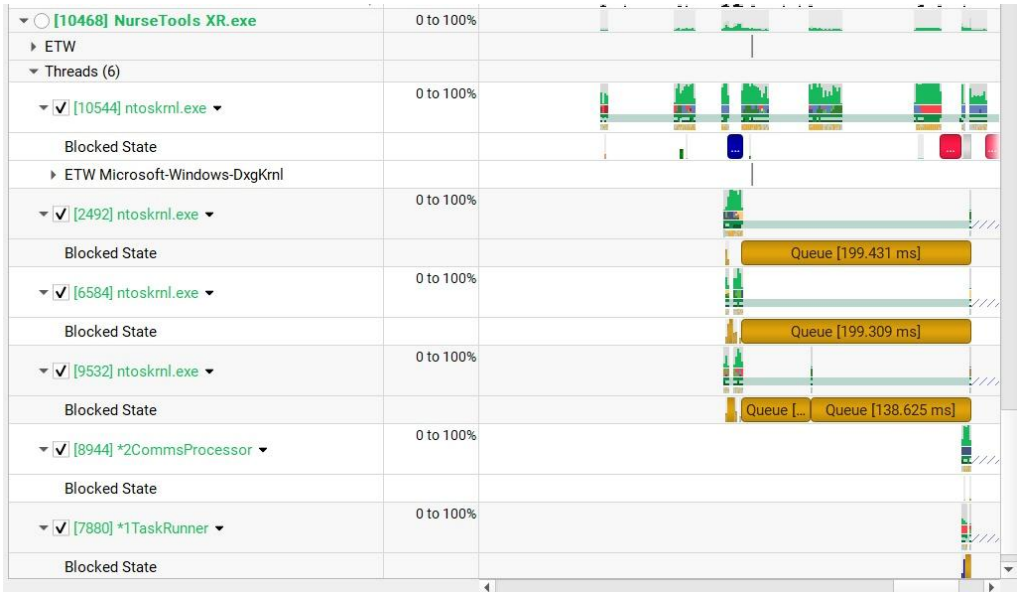


Figura 39:Procesos del sistema con colas críticas de GPU

De acuerdo con la Figura 39, expone procesos relacionados con el sistema (NurseTools XR.exe) y varios hilos del núcleo (ntoskrnl.exe) en estados de bloqueo, con colas que superan los 199 ms. Esto confirma que el rendimiento gráfico está siendo impactado negativamente por demoras prolongadas en la ejecución de tareas.

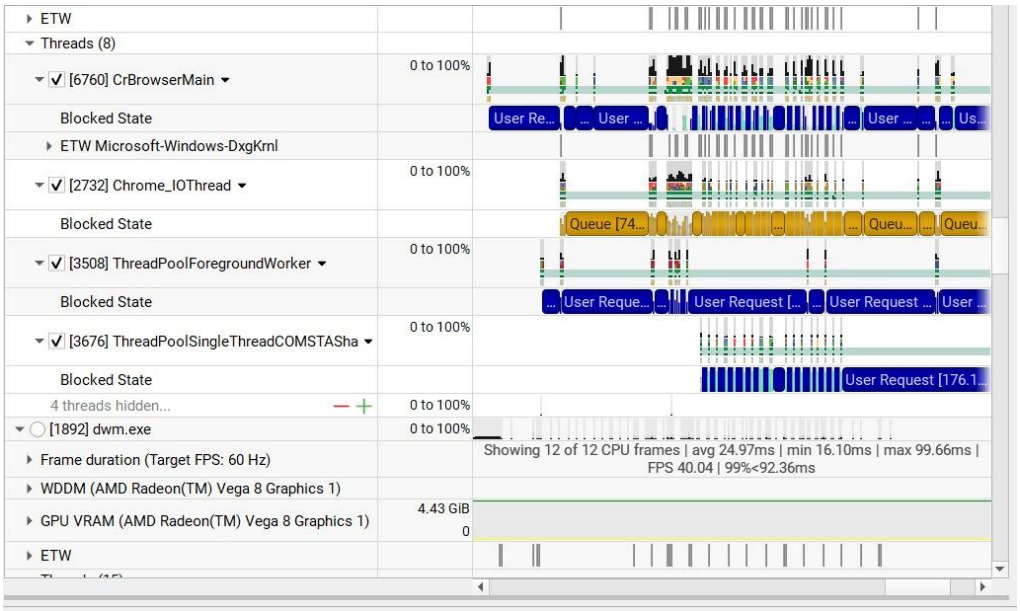


Figura 40: Análisis de Desempeño del Sistema mediante Nvidia Systems

La Figura 40, muestra una captura del análisis de trazas de rendimiento del sistema realizada mediante la herramienta Nvidia Systems (posiblemente Nsight Systems), enfocada en el comportamiento de múltiples hilos de procesos relacionados con el

navegador Chrome y el proceso del sistema dwm.exe (Desktop Window Manager). Se observan diversos hilos, entre ellos CrBrowserMain, Chrome_IOThread, y varios ThreadPool que presentan diferentes estados de ejecución y bloqueo, destacando numerosas solicitudes de usuario (etiquetadas como User Request) y colas de tareas (Queue).

En la parte inferior, se visualiza el consumo de VRAM de la GPU Radeon Vega 8, alcanzando los 4.43 GiB. También se presenta el rendimiento en cuadros por segundo (FPS), con un promedio de 40.04 FPS, un mínimo de 16.10 ms por cuadro, un máximo de 99.66 ms, y una eficiencia del 99% de cuadros por debajo de 92.36 ms, lo que indica un rendimiento general aceptable, aunque con ciertos picos de latencia.

Esta imagen evidencia cómo se distribuyen las cargas de trabajo entre los hilos del sistema y cómo estos interactúan con la GPU y el sistema operativo, ofreciendo una valiosa visión del comportamiento y desempeño general del entorno analizado.

3.6 Tiempo de Respuesta a Inputs (Experiencia de Usuario (UX))

Con el objetivo de evaluar la eficiencia y optimización del sistema, se registraron los tiempos de respuesta a distintos inputs dentro de la aplicación de realidad virtual desarrollada. Esta evaluación se realizó con la participación de 70 estudiantes, a quienes se les aplicaron dos mediciones: una durante la primera semana de uso y otra en la última semana del periodo de pruebas. Los tiempos promedio obtenidos permiten analizar cómo varió la rapidez de respuesta del sistema ante diversas interacciones, lo cual es un indicador importante del desempeño técnico de la aplicación y de la familiarización de los usuarios con la interfaz.

Análisis de resultados

Durante la primera semana de uso de la aplicación, los tiempos de respuesta a los distintos inputs fueron considerablemente variables, reflejando tanto el nivel inicial de familiarización de los estudiantes como posibles áreas de mejora en la optimización del sistema. Las acciones que requirieron más tiempo fueron la selección del tipo de operación (9.95 segundos) y la selección del tiempo de operación (12.98 segundos), lo cual sugiere una posible complejidad en la interfaz o en la comprensión de estas

funciones (ver Tabla 44). En contraste, tareas como cerrar la ventana de información (1.64 segundos) o ingresar al escenario (1.56 segundos) presentaron tiempos de respuesta significativamente menores, indicando que las interacciones más simples o directas eran más intuitivas para los usuarios. Estos resultados ofrecen una línea base importante para comparar el progreso en el uso de la aplicación y su rendimiento a lo largo del tiempo.

Tabla 44: Tiempo de respuesta a Inputs de la primera semana de prueba del sistema

Tiempo de respuesta a Inputs	**Promedio (Segundos)**
Entrar al escenario	1.56s
Iniciar la simulación	3.6s
Selección tipo de operación	9.95s
selección tiempo de operación	12.98s
Iniciar practica	3.21s
Ampliar instrumento	6.2s
Información instrumento	1.87s
Activar mesa mayo	3.97s
Seleccionar instrumento	3.53s
Iniciar simulación general	1.7s
Cerrar ventana información	1.64s

En la última semana de uso, se observaron mejoras notables en los tiempos de respuesta a la mayoría de los inputs, lo que refleja tanto una mayor familiarización de los estudiantes con la aplicación como una posible optimización en el rendimiento del sistema (ver Tabla 45). Las tareas que inicialmente presentaban mayores demoras, como la selección del tipo de operación y la selección del tiempo de operación, redujeron sus promedios a 4.51 y 5.23 segundos respectivamente, mostrando una disminución significativa respecto a la primera semana. Además, acciones básicas como iniciar la simulación (1.4 segundos), ingresar al escenario (1.16 segundos) y cerrar la ventana de información (1.64 segundos) mantuvieron o mejoraron sus tiempos, consolidando una experiencia de usuario más fluida y eficiente. Estos resultados evidencian una curva de aprendizaje positiva por parte de los usuarios y una mayor agilidad en la interacción con el entorno virtual.

Tabla 45: Tiempo de respuesta a Inputs de la cuarta semana de prueba del sistema

Tiempo de respuesta a Inputs	Promedio (Segundos)
Entrar al escenario	1.16s
Iniciar la simulación	1.4s
Selección tipo de operación	4.51s
selección tiempo de operación	5.23s
Iniciar practica	1.78s
Ampliar instrumento	3.3s
Información instrumento	1.8s
Activar mesa mayo	2.21s
Seleccionar instrumento	1.9s
Iniciar simulación general	1.21s
Cerrar ventana información	1.64s

Encuesta SUS para medir la usabilidad

Para evaluar la usabilidad de la aplicación, se utilizó la Escala de Usabilidad del Sistema (System Usability Scale, SUS), un cuestionario ampliamente reconocido y validado que permite medir de forma rápida y confiable la percepción de los usuarios sobre la facilidad de uso de un sistema. Este instrumento consta de diez ítems con una escala de tipo Likert de cinco puntos, que van desde "totalmente en desacuerdo" hasta "totalmente de acuerdo" (Anexo 3: Encuesta de Usabilidad). La SUS es especialmente útil en estudios donde se requiere obtener una visión general del desempeño de la interfaz desde la perspectiva del usuario, facilitando comparaciones y análisis cuantitativos de la experiencia de uso.

a. Aplicación y muestra

La aplicación fue puesta a prueba por una muestra de 70 estudiantes de medicina durante un periodo de cuatro semanas. Cada semana, los participantes utilizaban la aplicación de realidad virtual para simular prácticas clínicas específicas y, al finalizar cada sesión, respondían nuevamente el cuestionario SUS. Este enfoque permitió no solo medir la percepción inicial de usabilidad, sino también observar posibles variaciones en la experiencia del usuario a lo largo del tiempo, identificando mejoras o persistencia de dificultades en el uso de la aplicación. La recolección continua de datos facilitó un análisis evolutivo del nivel de aceptación y facilidad de uso percibida por los estudiantes.

b. Análisis de resultados

Pregunta 1: Me gustaría usar este sistema con frecuencia.

Semana 1

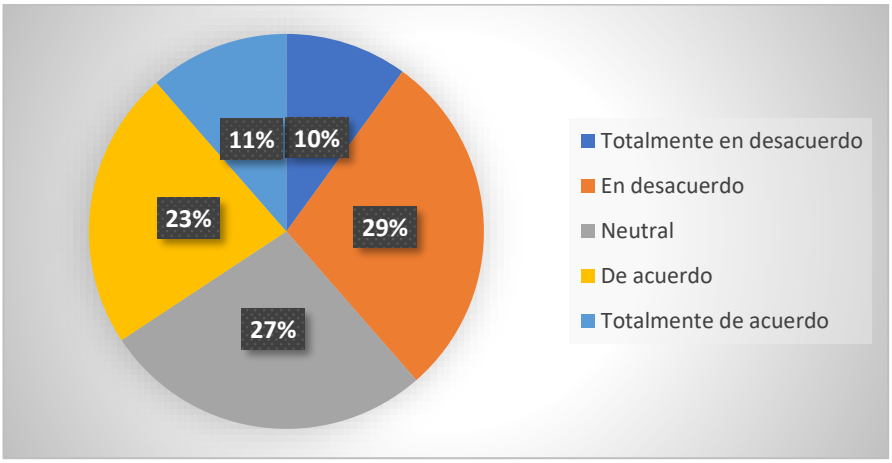


Figura 41: Frecuencia de resultados: pregunta 1, semana 1

Durante la primera semana, los resultados reflejaron una opinión bastante dividida entre los usuarios. Un 10% manifestó estar totalmente en desacuerdo con querer usar el sistema con frecuencia, mientras un 29% expresó estar en desacuerdo (ver Figura 41). Estos porcentajes suman un 39% de usuarios con una actitud negativa frente al sistema. Por otro lado, un 27% se mantuvo neutral, lo que puede interpretarse como indecisión o falta de suficiente interacción para emitir un juicio claro. Solo un 23% indicó estar de acuerdo y apenas un 11% manifestó estar totalmente de acuerdo. Esta distribución evidencia que, en su fase inicial, el sistema no generó un fuerte nivel de aceptación, posiblemente debido a factores como la novedad de la herramienta, una curva de aprendizaje no superada, o funcionalidades poco exploradas.

Semana 2

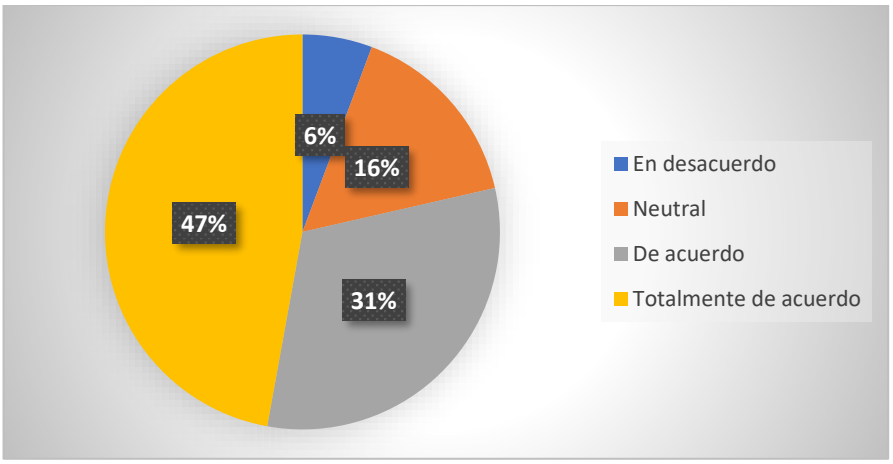


Figura 42: Frecuencia de resultados: pregunta 1, semana 2

En la segunda semana se observó una mejora considerable en la percepción general (ver Figura 42). El porcentaje de usuarios que expresó estar totalmente de acuerdo aumentó a un 47%, lo que representa casi la mitad de los encuestados. Además, un 31% se ubicó en la categoría de acuerdo, consolidando una mayoría de usuarios con opiniones favorables. No obstante, aún se mantuvieron niveles de escepticismo: un 16% se consideró neutral y un 6% aún se mostró en desacuerdo. La ausencia de usuarios en el nivel de totalmente en desacuerdo marca una mejora con respecto a la semana anterior. Este cambio puede interpretarse como una señal de adaptación y una experiencia de usuario más fluida, posiblemente gracias a una mayor familiaridad con el sistema y un entendimiento más claro de sus funcionalidades.

Semana 3

La evolución positiva continuó en la tercera semana. El 53% de los usuarios expresó estar totalmente de acuerdo con usar el sistema con frecuencia, mientras que el 40% indicó estar de acuerdo (ver Figura 43). Esto significa que un 93% de los encuestados manifestó una opinión favorable, consolidando una aceptación generalizada del sistema. Solo un 7% permaneció en una posición neutral, y ningún usuario expresó desacuerdo alguno. Este resultado sugiere que la mayoría de los usuarios ya se sienten cómodos y satisfechos con el sistema. La experiencia de uso parece haber mejorado

considerablemente, lo que podría deberse a una curva de aprendizaje ya superada y una interacción más eficiente con la plataforma.

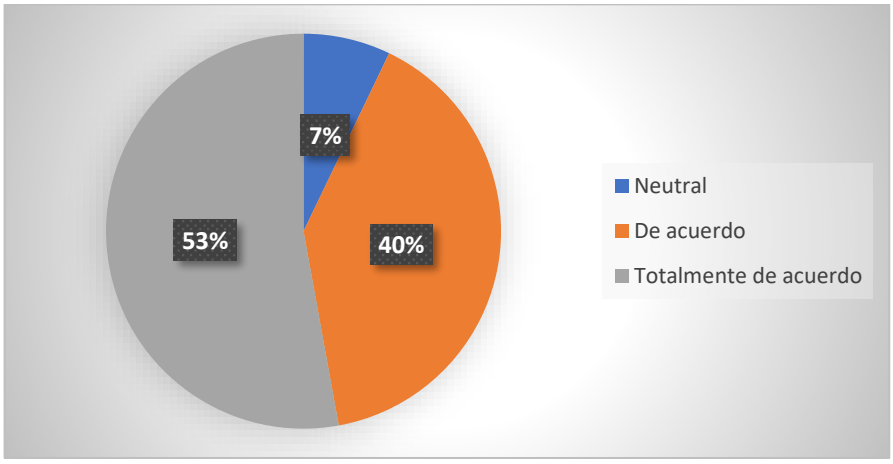


Figura 43: Frecuencia de resultados: pregunta 1, semana 3

Semana 4

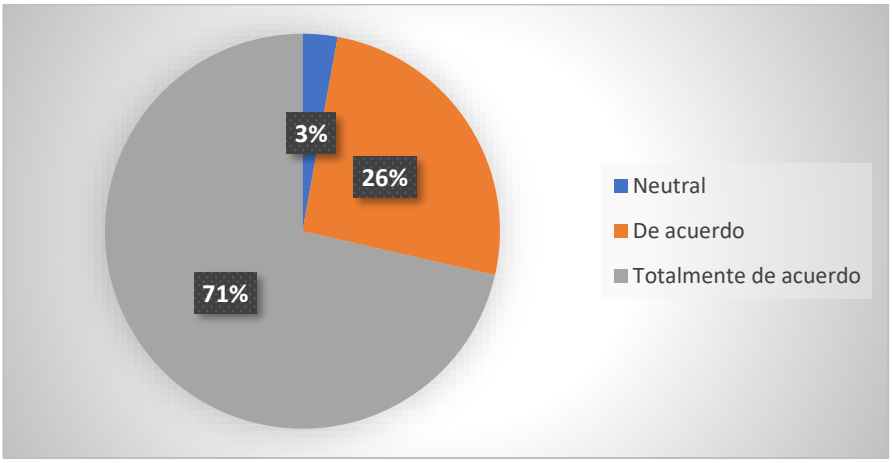


Figura 44: Frecuencia de resultados: pregunta 1, semana 4

En la última semana de medición, se alcanzaron los niveles más altos de aceptación (ver Figura 44). Un 71% de los participantes indicó estar totalmente de acuerdo y un 26% de acuerdo con la afirmación evaluada. Solo un 3% se mantuvo neutral, y nuevamente, no se presentaron respuestas negativas. Estos resultados demuestran que el sistema logró establecer una experiencia de usuario sólida y confiable. La evolución a lo largo del tiempo es clara: desde un punto inicial de duda y escepticismo, pasando por una etapa de adaptación, hasta llegar a una aceptación casi unánime. Esta

trayectoria sugiere que el sistema no solo es funcional, sino que logra generar un vínculo positivo con los usuarios una vez superadas las barreras iniciales.

Pregunta 2: Encontré el sistema innecesariamente complejo.

Semana 1

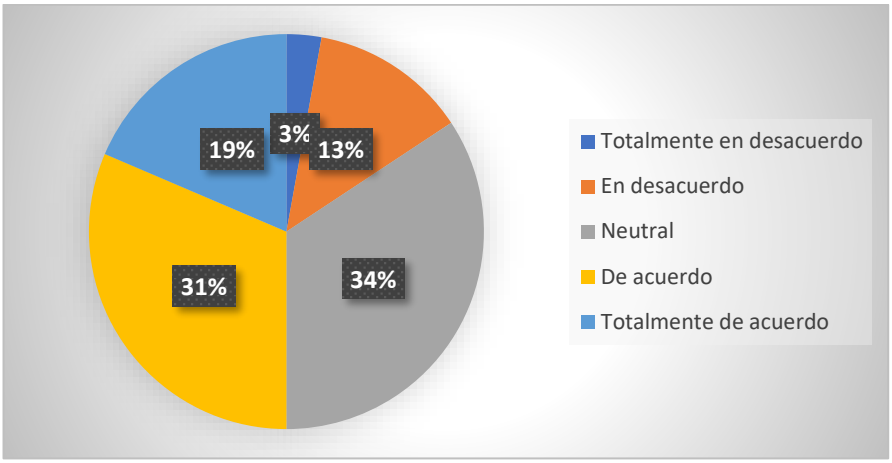


Figura 45: Frecuencia de resultados: pregunta 2, semana 1

En la primera semana, las percepciones estuvieron bastante distribuidas (ver Figura 45). Un 3% manifestó estar totalmente en desacuerdo con que el sistema fuera complejo, mientras un 13% estuvo en desacuerdo, lo cual indica que solo el 16% consideró el sistema sencillo desde el inicio. El grupo más amplio, el 34%, se posicionó como neutral, reflejando una percepción ambigua o falta de criterio formado. El 31% estuvo de acuerdo con que el sistema era innecesariamente complejo, y un preocupante 19% estuvo totalmente de acuerdo, mostrando que la mitad de los usuarios percibía cierta dificultad para interactuar con la herramienta. En resumen, hubo una fuerte impresión inicial de complejidad y una percepción no consolidada en los usuarios.

Semana 2

Durante la segunda semana, la percepción general mejoró. El grupo neutral creció hasta un 62%, lo cual sugiere que la mayoría de los usuarios aún no estaban completamente seguros sobre si el sistema era complejo o no, pero ya no lo consideraban tan difícil como antes ver (Figura 46). El porcentaje de usuarios en

desacuerdo con la afirmación (es decir, que no encontraban el sistema complejo) también aumentó a 23%. Además, solo el 4% se mantuvo totalmente en desacuerdo, y un 11% estuvo de acuerdo con la afirmación. Esta distribución indica una tendencia hacia la reducción de la percepción de complejidad, aunque todavía no se alcanzaba una opinión decididamente positiva.

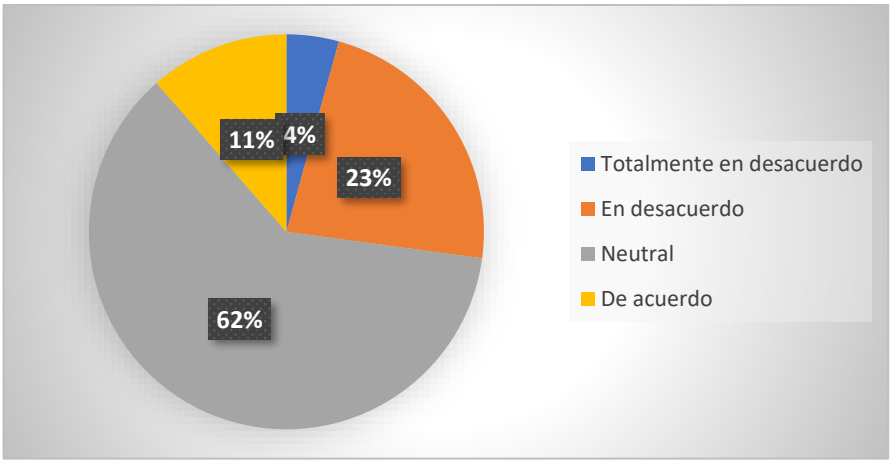


Figura 46: Frecuencia de resultados: pregunta 2, semana 2

Durante la segunda semana, la percepción general mejoró. El grupo neutral creció hasta un 62%, lo cual sugiere que la mayoría de los usuarios aún no estaban completamente seguros sobre si el sistema era complejo o no, pero ya no lo consideraban tan difícil como antes ver (Figura 46). El porcentaje de usuarios en desacuerdo con la afirmación (es decir, que no encontraban el sistema complejo) también aumentó a 23%. Además, solo el 4% se mantuvo totalmente en desacuerdo, y un 11% estuvo de acuerdo con la afirmación. Esta distribución indica una tendencia hacia la reducción de la percepción de complejidad, aunque todavía no se alcanzaba una opinión decididamente positiva.

Semana 3

En la tercera semana, el cambio fue más notorio (ver Figura 47). El 51% de los encuestados se mantuvo neutral, lo que sigue evidenciando una postura ambigua pero menos negativa. Solo un 30% estuvo en desacuerdo con que el sistema fuera complejo, y un 10% se mantuvo en la categoría totalmente en desacuerdo, lo cual representa un crecimiento del reconocimiento positivo. En contraste, apenas un 9% todavía estuvo

de acuerdo con la afirmación. Esta distribución muestra que la percepción de complejidad seguía disminuyendo y que los usuarios ya comenzaban a considerar el sistema más accesible, probablemente como resultado de un uso más frecuente y familiarización progresiva.

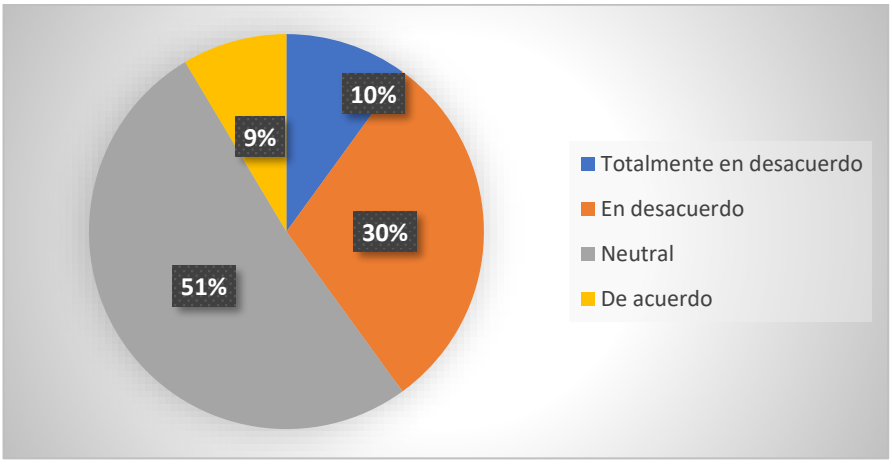


Figura 47: Frecuencia de resultados: pregunta 2, semana 3

Semana 4

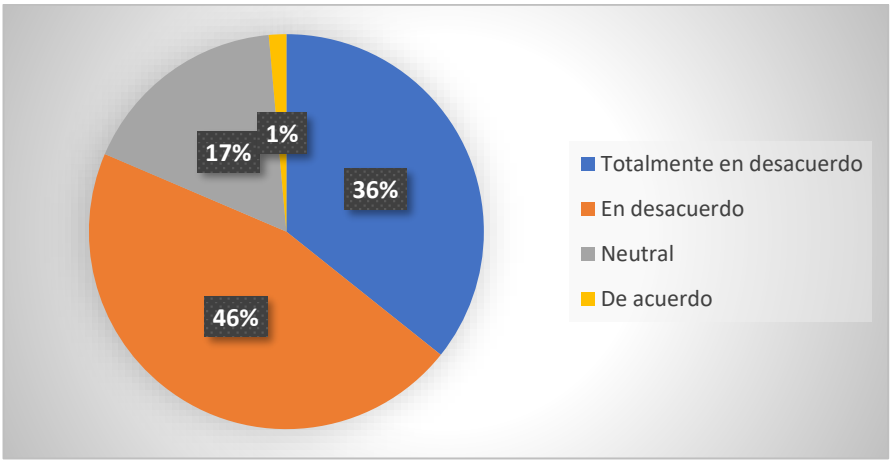


Figura 48: Frecuencia de resultados: pregunta 2, semana 4

En la última semana, sorprendentemente, se produjo un retroceso en la percepción positiva (ver Figura 48). Un 46% de los encuestados expresó estar en desacuerdo con la afirmación, lo cual es positivo, pero un 36% manifestó estar totalmente en desacuerdo, una mejora significativa con respecto a las semanas anteriores. Sin embargo, el dato más llamativo es que solo un 1% se mantuvo neutral, y un 17%

declaró estar de acuerdo con que el sistema era innecesariamente complejo. Aunque el porcentaje negativo disminuyó en comparación con la semana 1, la ligera reaparición de valoraciones negativas sugiere que, si bien la mayoría ya no encontraba el sistema complejo, aún existen áreas donde podría simplificarse o mejorarse para lograr una experiencia uniforme entre todos los usuarios.

Pregunta 3: Consideré que el sistema era fácil de usar.

Semana 1

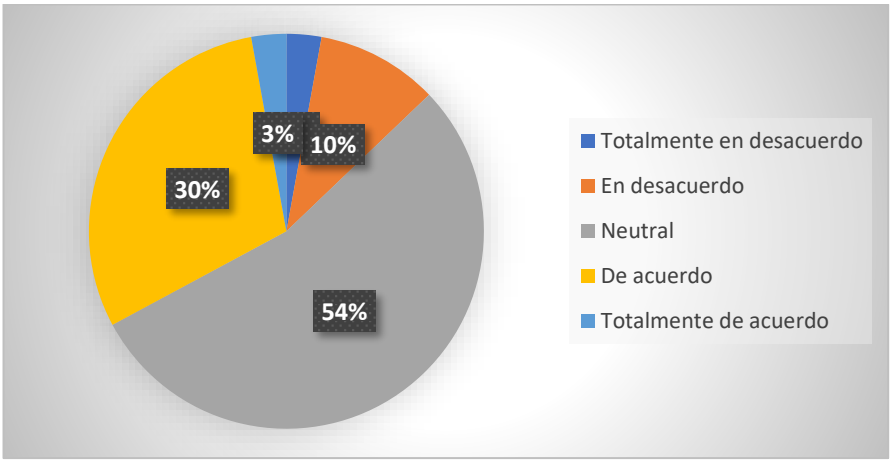


Figura 49: Frecuencia de resultados: pregunta 3, semana 1

En la primera semana, los resultados fueron marcadamente positivos. El 54% de los encuestados expresó estar de acuerdo con que el sistema era fácil de usar, y un 30% estuvo totalmente de acuerdo, lo que representa un 84% de aceptación directa de la facilidad de uso (ver Figura 49). En contraste, solo un 10% se mostró neutral, y el 3% tanto en desacuerdo como totalmente en desacuerdo. Estos porcentajes sugieren que, desde el inicio, una mayoría de usuarios percibió el sistema como accesible y comprensible, lo cual es un indicador positivo sobre la usabilidad en su fase temprana.

Semana 2

Durante la segunda semana se observó una ligera disminución en las opiniones más favorables. El porcentaje de usuarios *de acuerdo* cayó al 34%, y quienes se mostraron *totalmente de acuerdo* se redujeron al 27% (ver Figura 50). En cambio, el grupo *neutral* creció hasta el 32%, reflejando una percepción más ambigua. Un 7% expresó

estar en desacuerdo, sin presencia de respuestas *totalmente en desacuerdo*. Este cambio puede interpretarse como resultado de una exposición más crítica o detallada al sistema, donde algunos usuarios empezaron a notar aspectos mejorables, aunque la mayoría seguía considerando el sistema fácil de usar.

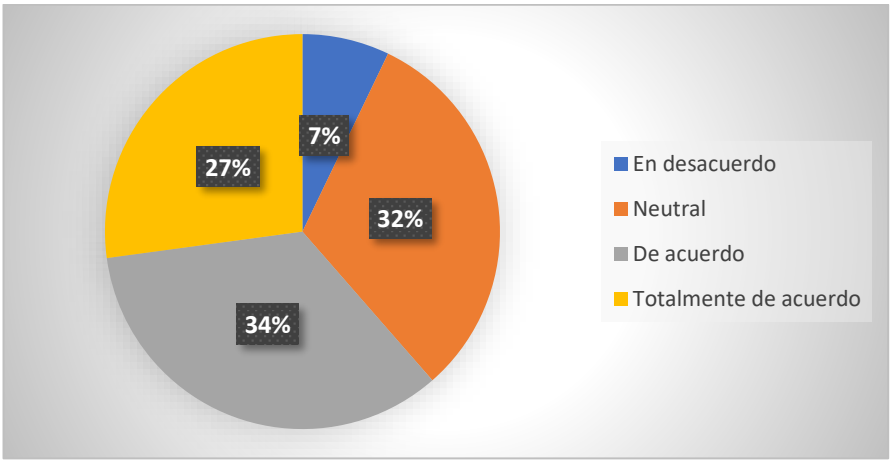


Figura 50: Frecuencia de resultados: pregunta 3, semana 2

Semana 3

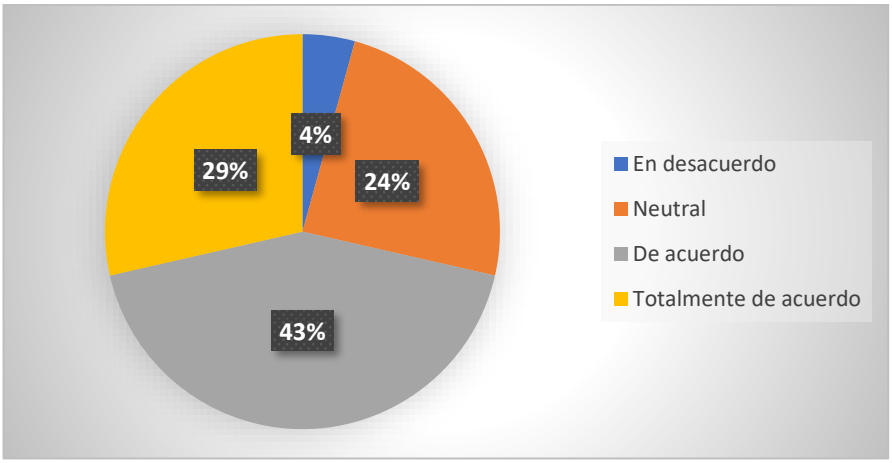


Figura 51: Frecuencia de resultados: pregunta 3, semana 3

En la tercera semana se estabilizó la percepción positiva (ver Figura 51). Un 43% declaró estar de acuerdo y un 29% totalmente de acuerdo, representando un 72% de valoración favorable. El grupo neutral fue del 24%, mientras que solo un 4% manifestó estar en desacuerdo. Esta distribución refleja que la mayoría de los usuarios mantenía una opinión positiva, y aunque hubo una ligera reducción frente a la primera semana,

la percepción general seguía siendo optimista. La caída en las puntuaciones más altas puede deberse a exigencias mayores por parte de los usuarios a medida que profundizan en el uso del sistema.

Semana 4

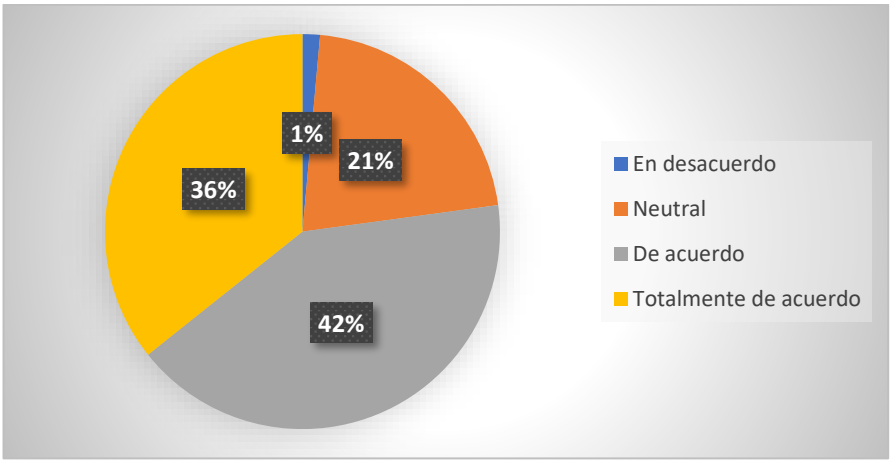


Figura 52: Frecuencia de resultados: pregunta 3, semana 4

En la cuarta semana se recuperaron nuevamente niveles altos de aceptación. Un 42% manifestó estar de acuerdo con la afirmación, y un significativo 36% estuvo totalmente de acuerdo. Solo un 21% se mantuvo neutral, mientras que apenas un 1% expresó estar en desacuerdo (ver Figura 52). Esta evolución final refuerza la idea de que, conforme avanza el tiempo, el sistema es percibido con mayor facilidad de uso. Los usuarios parecen haber superado posibles dificultades iniciales, lo que evidencia una mejora en la experiencia general de interacción y una consolidación del diseño en términos de usabilidad.

Pregunta 4: Creo que necesitaría la ayuda de un experto para usar este sistema.

Semana 1

En la primera semana, los resultados muestran una percepción moderadamente favorable (ver Figura 53). Un 37% de los usuarios expresó estar en desacuerdo con necesitar ayuda experta, mientras que un 17% estuvo totalmente de acuerdo con la afirmación, revelando una posible inseguridad inicial entre ciertos usuarios. Un 29% se mantuvo neutral, reflejando incertidumbre. Solo un 1% estuvo totalmente en

desacuerdo, lo que sugiere que muy pocos se sentían completamente autosuficientes desde el inicio. La distribución evidencia que, si bien una parte importante del grupo encontraba el sistema manejable, aún había quienes no se sentían totalmente autónomos.

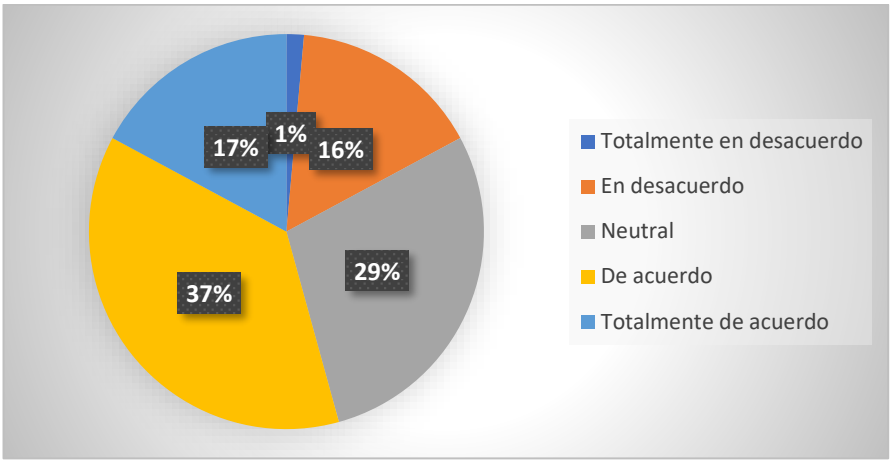


Figura 53: Frecuencia de resultados: pregunta 4, semana 1

Semana 2

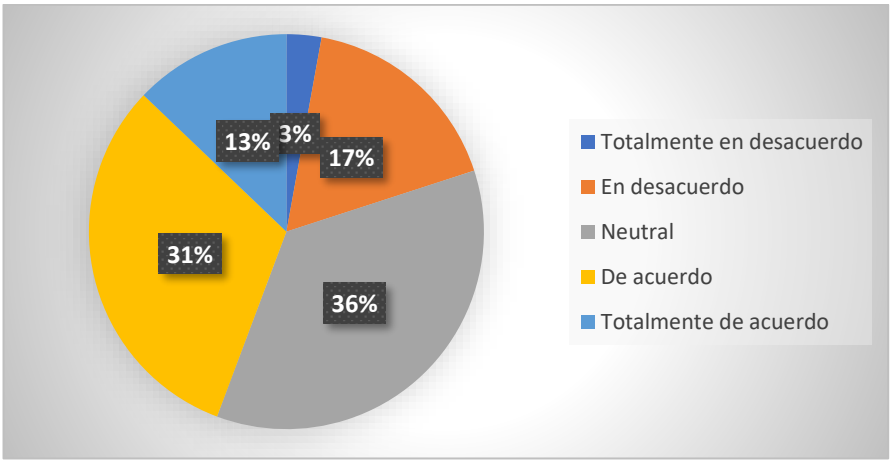


Figura 54: Frecuencia de resultados: pregunta 4, semana 2

Durante la segunda semana, los usuarios continuaron mostrando un nivel equilibrado de percepciones (ver Figura 54). El porcentaje de usuarios en desacuerdo aumentó levemente al 31%, mientras que los neutrales se ubicaron en 36%. Las respuestas negativas se redujeron: un 17% estuvo en desacuerdo con la afirmación (positivo), y un 13% declaró estar totalmente de acuerdo, evidenciando una leve disminución en la

percepción de necesidad de asistencia. Aunque no hubo cambios drásticos, la tendencia apuntaba a una progresiva familiarización con el sistema, aunque seguía presente un grupo que requería guía.

Semana 3

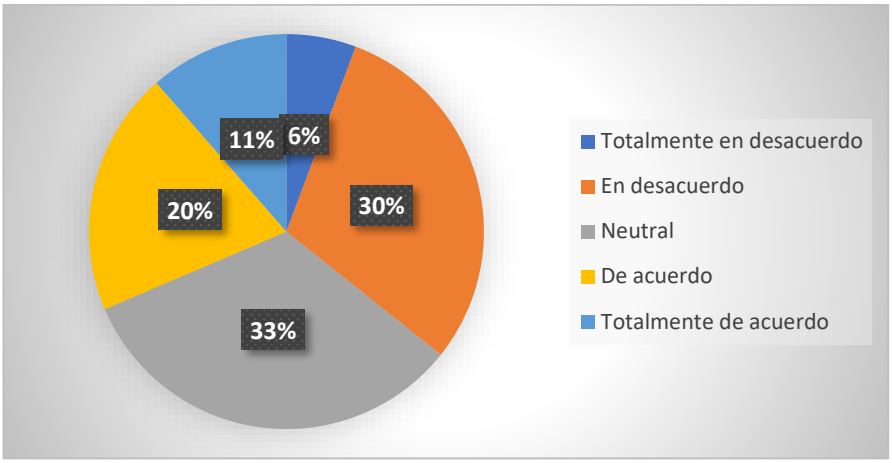


Figura 55: Frecuencia de resultados: pregunta 4, semana 3

En esta semana, la distribución fue bastante equitativa (ver Figura 55). Un 33% permaneció neutral, indicando una estabilidad en la percepción de autoeficacia. Un 30% expresó estar en desacuerdo, mientras que un 20% estuvo de acuerdo y un 11% incluso totalmente de acuerdo, lo que sugiere que, si bien un grupo notable seguía sintiendo la necesidad de apoyo, la mayoría comenzaba a desenvolverse con mayor confianza. La percepción se mantuvo en un punto medio sin progresos muy marcados, reflejando que aún existían barreras de dominio para ciertos usuarios.

Semana 4

En la cuarta semana, se produjo un cambio importante (ver Figura 56): el 54% manifestó estar en desacuerdo con la necesidad de ayuda experta, el porcentaje más alto hasta el momento. Además, el 32% se mantuvo neutral, lo que deja ver que una mayoría sentía autonomía o al menos ya no requería asistencia constante. Solo un 10% indicó estar totalmente en desacuerdo (positivo), y un mínimo 4% declaró estar de acuerdo, lo que representa un retroceso claro en la percepción de dependencia. Esta mejora sugiere que los usuarios han ganado experiencia suficiente para operar el

sistema con confianza, reflejando una evolución favorable en la facilidad de uso percibida y la independencia operativa.

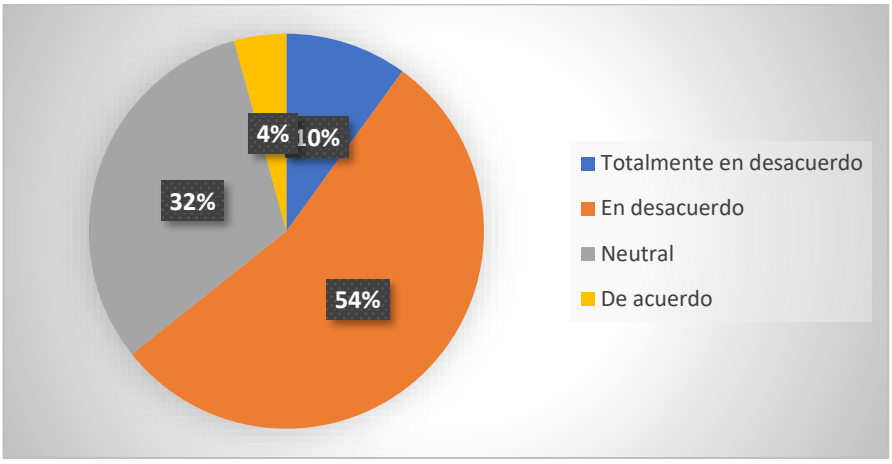


Figura 56: Frecuencia de resultados: pregunta 4, semana 4

Pregunta 5: Las funciones del sistema estaban bien integradas.

Semana 1

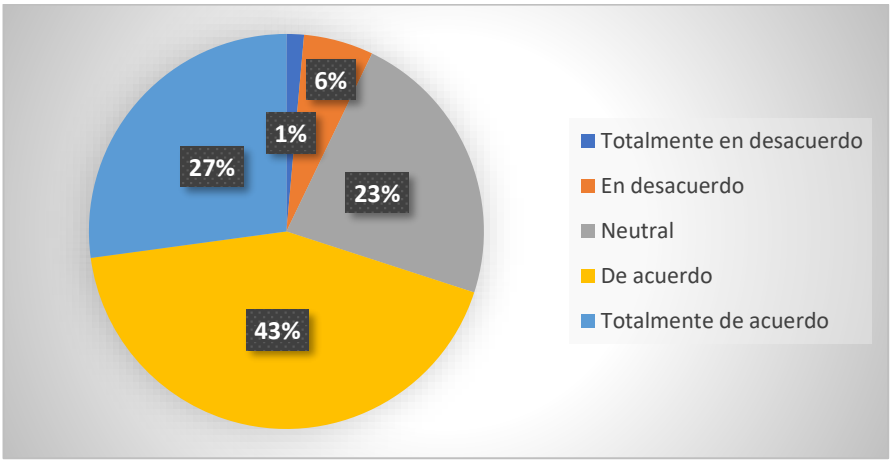


Figura 57: Frecuencia de resultados: pregunta 5, semana 1

En esta semana inicial, la mayoría de los usuarios (43%) indicó estar de acuerdo con que las funciones del sistema estaban bien integradas (ver Figura 57). Un 27% adicional estuvo totalmente de acuerdo, lo que sugiere que más de dos tercios de los participantes percibieron una integración funcional aceptable desde el comienzo. Sin embargo, un 23% se mantuvo neutral, lo que podría interpretarse como una falta de certeza al respecto. Solo un 6% estuvo en desacuerdo y un 1% totalmente en

desacuerdo, indicando que las críticas a la integración eran mínimas. En conjunto, se trató de una valoración inicial bastante prometedora.

Semana 2

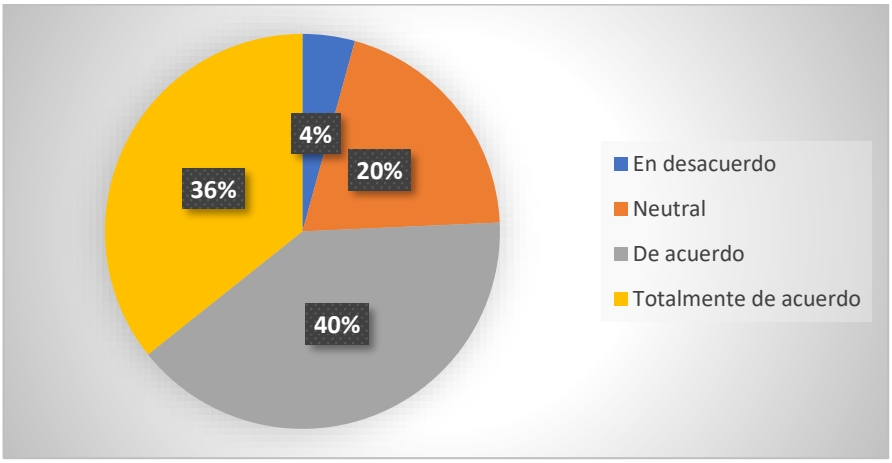


Figura 58: Frecuencia de resultados: pregunta 5, semana 2

Durante la segunda semana, la tendencia positiva se mantuvo estable (ver Figura 58). El 40% de los usuarios manifestó estar de acuerdo con la afirmación, y un 36% estuvo totalmente de acuerdo, alcanzando un 76% de percepción positiva. Un 20% se mantuvo neutral, y solo un 4% expresó estar en desacuerdo. Estos resultados muestran una ligera mejora en la confianza sobre la integración del sistema. Esto puede explicarse por un uso más profundo del sistema, que permitió a los usuarios identificar una estructura funcional más cohesionada.

Semana 3

En esta semana, se observó una consolidación de la percepción positiva. El 38% expresó estar de acuerdo, mientras que un significativo 43% estuvo totalmente de acuerdo (ver Figura 59). Solo un 16% se ubicó en la categoría neutral y un escaso 3% en desacuerdo. Esta distribución sugiere que, para la mayoría, la funcionalidad del sistema se percibía como fluida y coherente. A estas alturas, la interacción constante con la herramienta parece haber reforzado la apreciación de su diseño estructural.

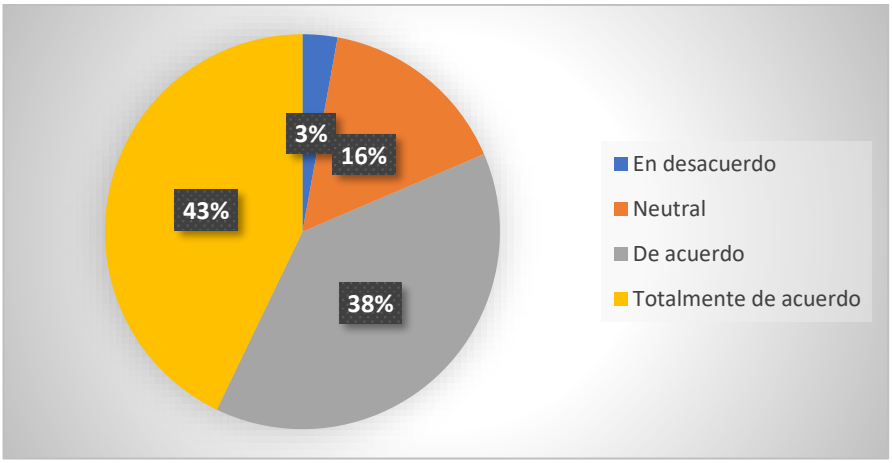


Figura 59: Frecuencia de resultados: pregunta 5, semana 3

Semana 4

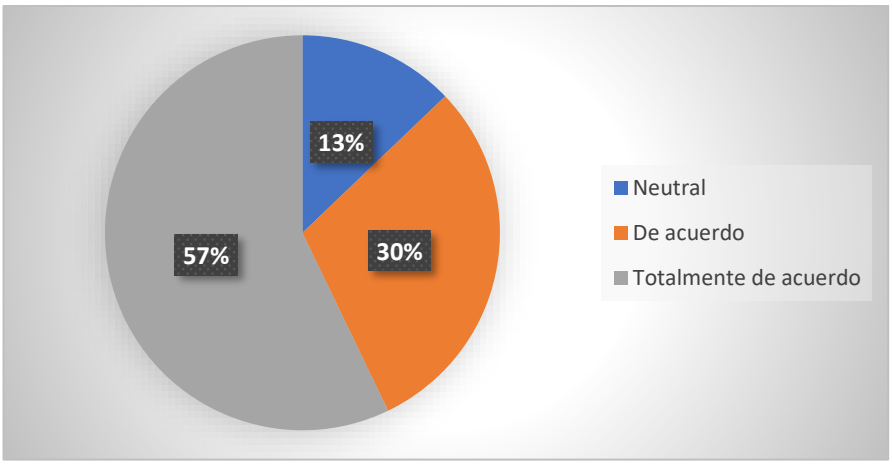


Figura 60: Frecuencia de resultados: pregunta 5, semana 4

En la última semana, los resultados fueron los más contundentes. Un 57% de los usuarios manifestó estar totalmente de acuerdo con que las funciones estaban bien integradas, y un 30% estuvo de acuerdo, lo que da un impresionante 87% de aprobación (ver Figura 60). Solo un 13% se mantuvo neutral y no hubo respuestas negativas. Esta valoración final refleja una percepción sólida de que el sistema, tras varias semanas de uso, fue entendido por los usuarios como una herramienta con componentes bien coordinados y lógica interna clara. La evolución muestra una mejora sostenida que culmina en una percepción altamente satisfactoria sobre su integración funcional.

Pregunta 6: El sistema era inconsistente.

Semana 1

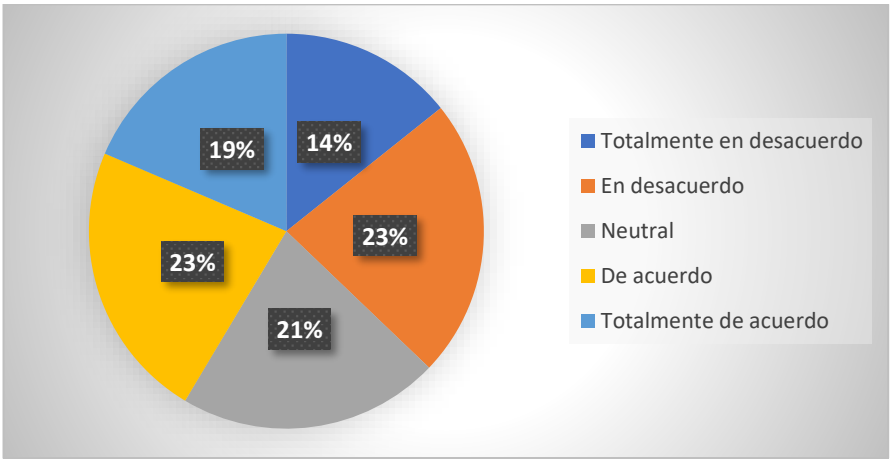


Figura 61: Frecuencia de resultados: pregunta 6, semana 1

Durante la primera semana, las respuestas estuvieron bastante distribuidas. Un 23% de los usuarios manifestó estar en desacuerdo y otro 14% totalmente en desacuerdo con la afirmación, lo cual es positivo, ya que sugiere que más de un tercio de los encuestados no percibió inconsistencias en el sistema (ver Figura 61). Sin embargo, también hubo un 21% neutral, un 23% de acuerdo y un 19% totalmente de acuerdo, lo cual representa casi la mitad del grupo con una percepción negativa o ambigua. Esto refleja que, al inicio, hubo dudas importantes sobre la coherencia funcional del sistema, probablemente producto del desconocimiento inicial o de una interfaz poco clara.

Semana 2

En esta semana, el porcentaje de usuarios que consideró al sistema inconsistente aumentó: un 46% estuvo de acuerdo con la afirmación y un 8% totalmente de acuerdo. Esto representa una intensificación en la percepción negativa (ver Figura 62). Solo un 19% estuvo totalmente en desacuerdo, mientras que un 27% se mantuvo neutral. El aumento en el desacuerdo parcial y total podría estar vinculado a una mayor exploración del sistema, que reveló a los usuarios ciertas incoherencias entre funcionalidades o comportamientos inesperados.

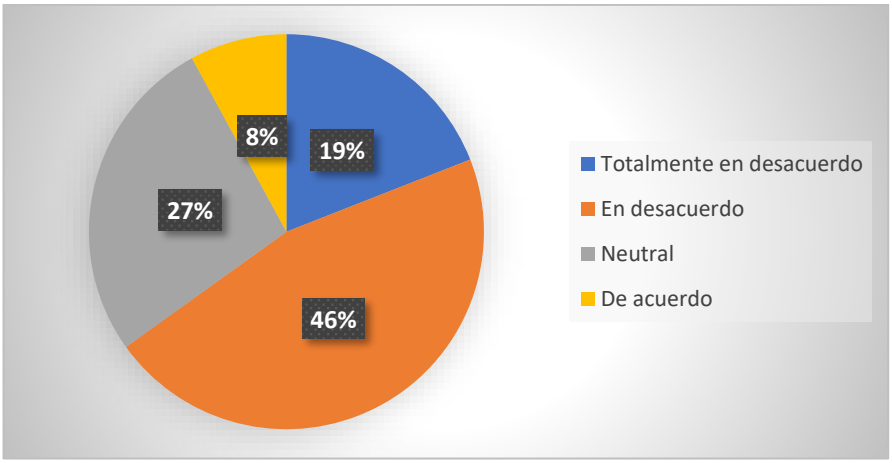


Figura 62: Frecuencia de resultados: pregunta 6, semana 2

Semana 3

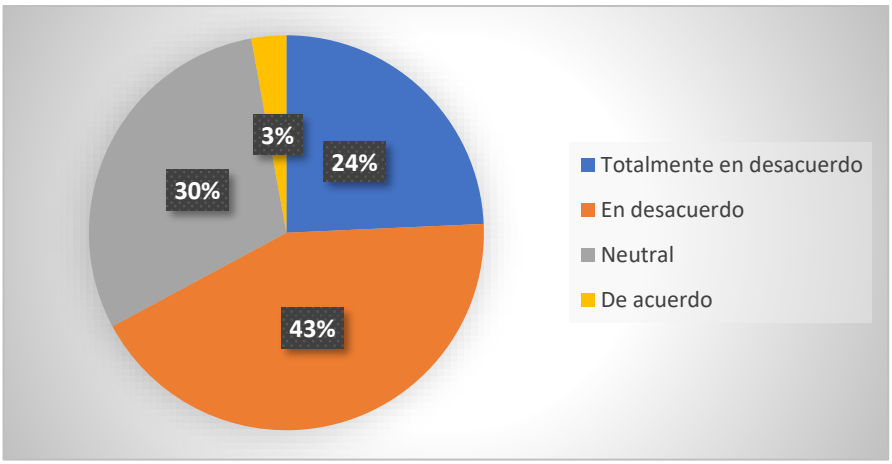


Figura 63: Frecuencia de resultados: pregunta 6, semana 3

La percepción negativa se mantuvo alta: un 43% se mostró de acuerdo con que el sistema era inconsistente y un 3% totalmente de acuerdo. Por otro lado, solo un 24% estuvo totalmente en desacuerdo, y un 30% se mantuvo neutral. Estos datos indican que, si bien no hubo un aumento en las críticas, tampoco se logró reducir de forma significativa la percepción de inconsistencia. Esta tendencia sugiere que algunos problemas percibidos en semanas anteriores persistieron sin mejoras notables desde la perspectiva del usuario (ver Figura 63).

Semana 4

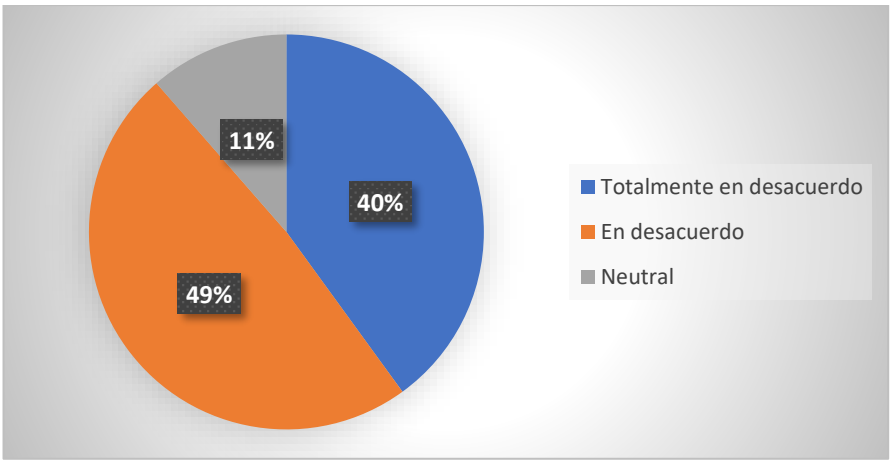


Figura 64: Frecuencia de resultados: pregunta 6, semana 4

En la última semana, se presentó una mejora considerable (ver Figura 64). El 40% declaró estar totalmente en desacuerdo y el 49% en desacuerdo con que el sistema fuera inconsistente, lo que representa un 89% de evaluación positiva. Solo un 11% se mantuvo neutral, y ningún usuario se expresó a favor de la afirmación. Esto representa una evolución clara y favorable, donde la percepción de consistencia del sistema aumentó significativamente. Los usuarios, al final del período, reconocieron que el sistema funcionaba de manera coherente, lo que refleja mejoras en la interfaz, comprensión de uso, o simplemente adaptación del usuario.

Pregunta 7: Creo que la mayoría de las personas aprenderían a usar este sistema muy rápidamente.

Semana 1

En la primera semana, las opiniones estuvieron distribuidas de manera bastante equilibrada (ver Figura 65). Un 27% de los usuarios estuvo de acuerdo con la afirmación, y un 10% totalmente de acuerdo, sumando un 37% de percepción positiva sobre la facilidad de aprendizaje del sistema. Sin embargo, también hubo un 24% de usuarios neutrales, y otro 24% expresó estar en desacuerdo, lo cual indica cierto nivel de escepticismo. Un 15% incluso estuvo totalmente en desacuerdo, reflejando que un segmento significativo no consideraba que el sistema fuera fácilmente comprensible para nuevos usuarios. Estos resultados reflejan un punto de partida donde la

experiencia de uso aún no convencía completamente en términos de accesibilidad para otros.

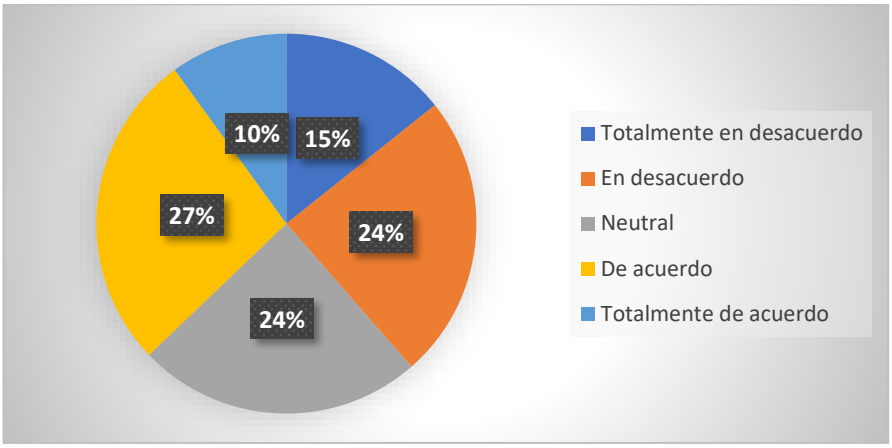


Figura 65: Frecuencia de resultados: pregunta 7, semana 1

Semana 2

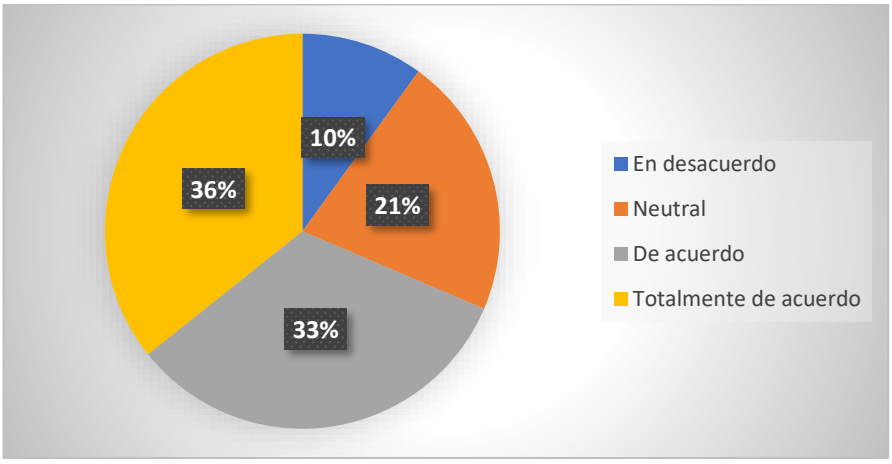


Figura 66: Frecuencia de resultados: pregunta 7, semana 2

Durante la segunda semana, las percepciones positivas mejoraron ligeramente (ver Figura 66). Un 33% de los encuestados estuvo de acuerdo con que el sistema podía aprenderse rápidamente, mientras que un 36% se mostró totalmente de acuerdo, totalizando un 69% de aceptación. El porcentaje de usuarios neutrales fue del 21%, y un 10% se mantuvo en desacuerdo, sin presencia de respuestas totalmente negativas. Esto muestra que a medida que los usuarios continuaban usando el sistema, la confianza en su facilidad de aprendizaje aumentaba de forma notoria.

Semana 3

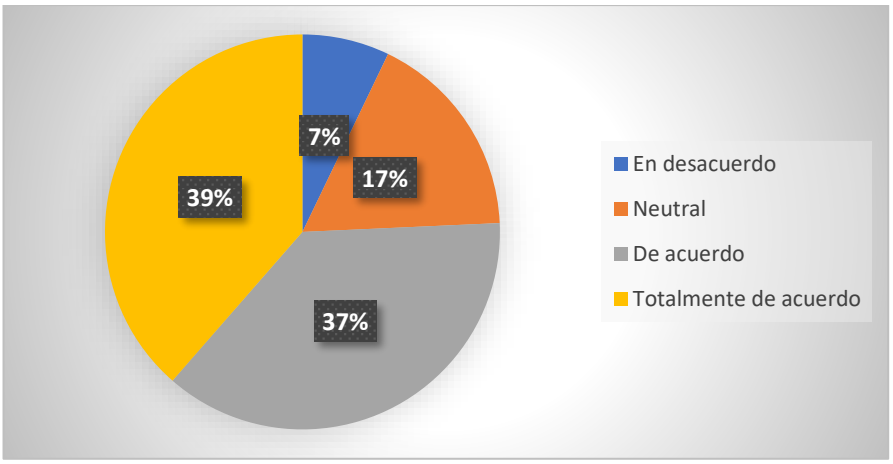


Figura 67: Frecuencia de resultados: pregunta 7, semana 3

En esta semana, la tendencia favorable se mantuvo (ver Figura 67). Un 37% de los usuarios estuvo de acuerdo y un 39% totalmente de acuerdo, consolidando un 76% de percepción positiva. El grupo neutral disminuyó al 17%, y solo un 7% expresó estar en desacuerdo. Esta evolución demuestra una clara mejora en la percepción colectiva: los usuarios no solo entendieron el sistema con mayor profundidad, sino que comenzaron a confiar en que otras personas también podrían dominarlo rápidamente.

Semana 4

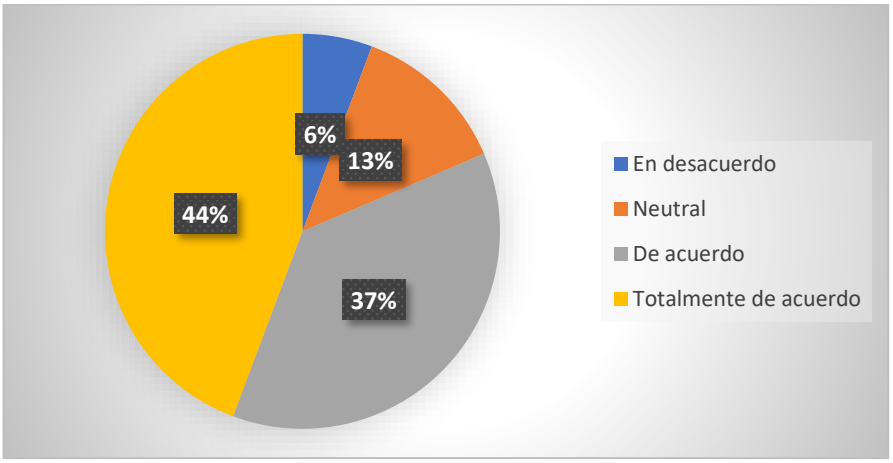


Figura 68: Frecuencia de resultados: pregunta 7, semana 4

En la última semana (ver Figura 68), los resultados continuaron siendo positivos. El 44% de los participantes manifestó estar totalmente de acuerdo con la afirmación, y

un 37% estuvo de acuerdo, acumulando un sólido 81% de opinión favorable. Solo un 13% permaneció neutral y un 6% expresó en desacuerdo. Estos valores reflejan que, al finalizar el proceso de uso, la percepción de facilidad de aprendizaje fue ampliamente aceptada por la mayoría. El sistema no solo resultó accesible para quienes lo utilizaron, sino que también fue percibido como una herramienta que cualquier nuevo usuario podría comprender sin dificultad.

Pregunta 8: Consideré que el sistema fue difícil de usar.

Semana 1

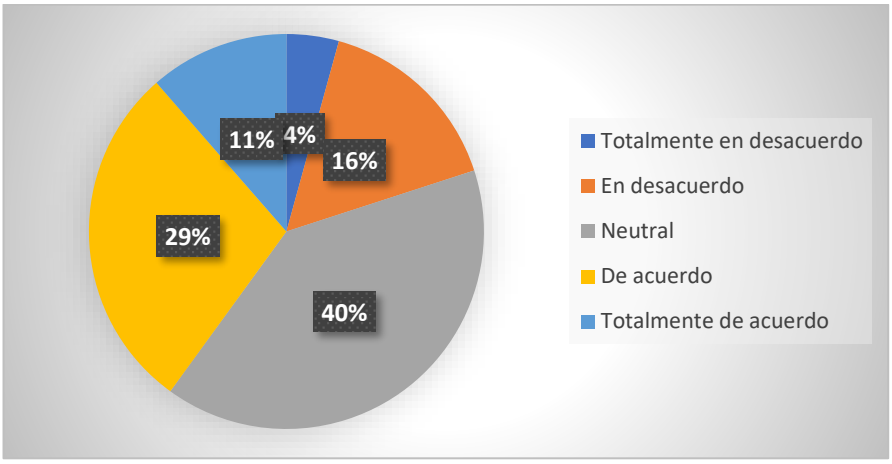


Figura 69: Frecuencia de resultados: pregunta 8, semana 1

En esta primera semana, el 40% de los usuarios estuvo neutral frente a la afirmación, lo que refleja incertidumbre o una falta de criterio sólido sobre la dificultad del sistema (ver Figura 69). Un 29% expresó estar totalmente en desacuerdo y un 11% en desacuerdo, lo que suma un 40% con percepción positiva. Por otro lado, un 16% se mostró de acuerdo y un 4% totalmente de acuerdo, lo que representa una proporción menor de usuarios que encontraron el sistema difícil de usar. Estos resultados iniciales indican una experiencia de uso razonablemente aceptable, aunque con una amplia porción de usuarios aún indecisos.

Semana 2

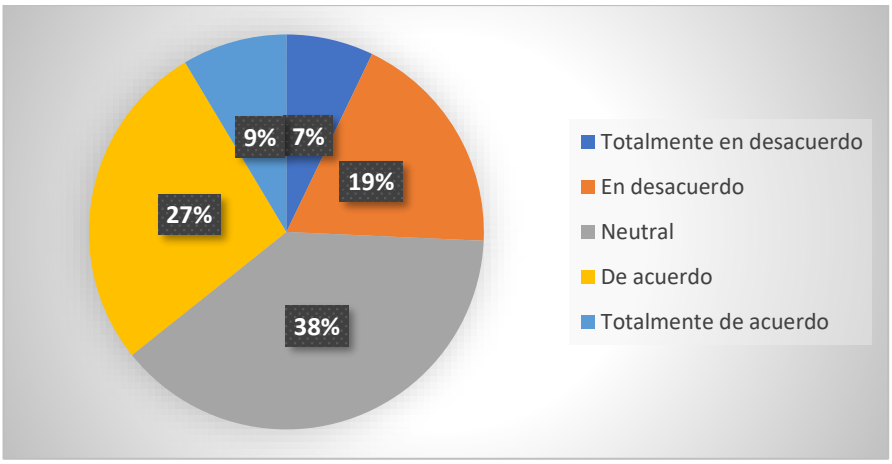


Figura 70: Frecuencia de resultados: pregunta 8, semana 2

Durante la segunda semana, el porcentaje de usuarios con percepción positiva disminuyó ligeramente (ver Figura 70). Solo el 27% estuvo totalmente en desacuerdo, mientras que el 19% estuvo en desacuerdo. Sin embargo, la categoría neutral creció hasta el 38%, lo que muestra que más personas adoptaron una postura ambigua respecto a la facilidad de uso. A su vez, el 7% estuvo de acuerdo y el 9% totalmente de acuerdo, lo que sugiere una leve inclinación hacia percepciones negativas. En resumen, esta semana mostró un retroceso en la claridad de la experiencia: muchos usuarios aún no se sentían completamente cómodos con el sistema.

Semana 3

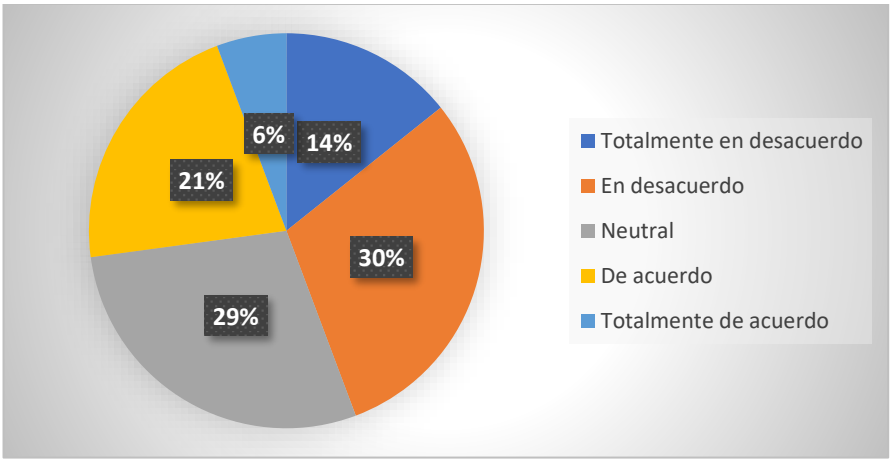


Figura 71: Frecuencia de resultados: pregunta 8, semana 3

En la tercera semana, la distribución fue más balanceada. Un 21% expresó estar totalmente en desacuerdo y un 14% en desacuerdo, sumando un 35% de percepción positiva (ver Figura 71). Un 29% se mantuvo neutral, mientras que el 30% se mostró de acuerdo con que el sistema era difícil de usar y un 6% totalmente de acuerdo. Aunque las percepciones positivas y negativas se mantuvieron cerca, se nota que la percepción de dificultad seguía siendo un tema para una porción importante de los usuarios, probablemente aquellos menos habituados al tipo de interfaz utilizada.

Semana 4

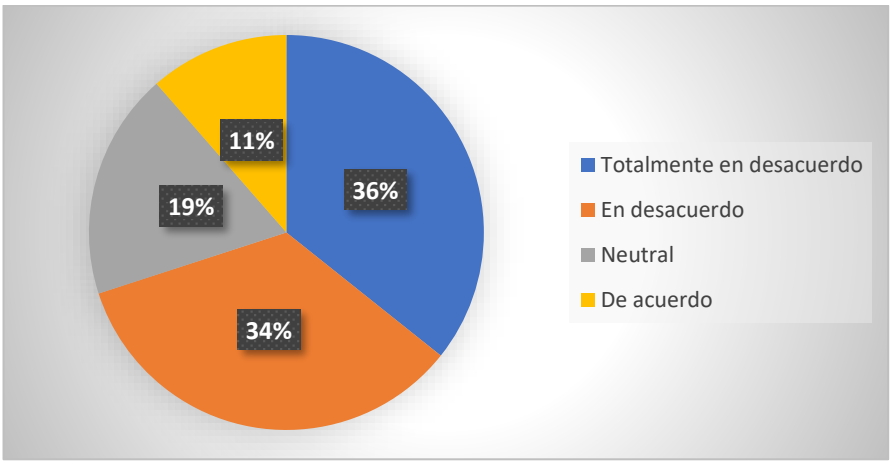


Figura 72: Frecuencia de resultados: pregunta 8, semana 4

En la última semana se registró una mejora sustancial en la percepción de facilidad de uso (ver Figura 72). Un 36% de los encuestados expresó estar totalmente en desacuerdo y un 34% en desacuerdo con que el sistema fuera difícil de usar, sumando un 70% de valoración positiva. Solo un 19% permaneció neutral y un 11% expresó algún nivel de acuerdo. Esta mejora es significativa y sugiere que, con el paso del tiempo y la familiarización, la mayoría de los usuarios terminó considerando el sistema como fácil de usar, con una reducción visible en las opiniones negativas.

Pregunta 9: Me sentí seguro usando este sistema.

Semana 1

Durante la primera semana, las respuestas estuvieron divididas. El 41% de los usuarios manifestó estar de acuerdo con sentirse seguros usando el sistema, mientras que un 9%

estuvo totalmente de acuerdo (ver Figura 73). Por otro lado, el 40% permaneció neutral, lo que indica una falta de convicción o una experiencia aún insuficiente para emitir un juicio claro. Solo un 10% expresó estar en desacuerdo. Este escenario inicial refleja que, aunque existía una mayoría que empezaba a sentirse cómoda, aún quedaba una gran parte que no terminaba de confiar plenamente en el sistema.

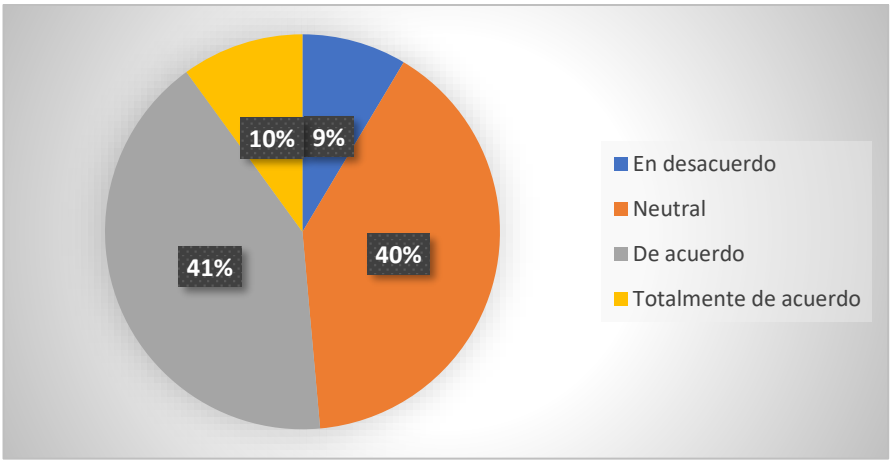


Figura 73: Frecuencia de resultados: pregunta 9, semana 1

Semana 2

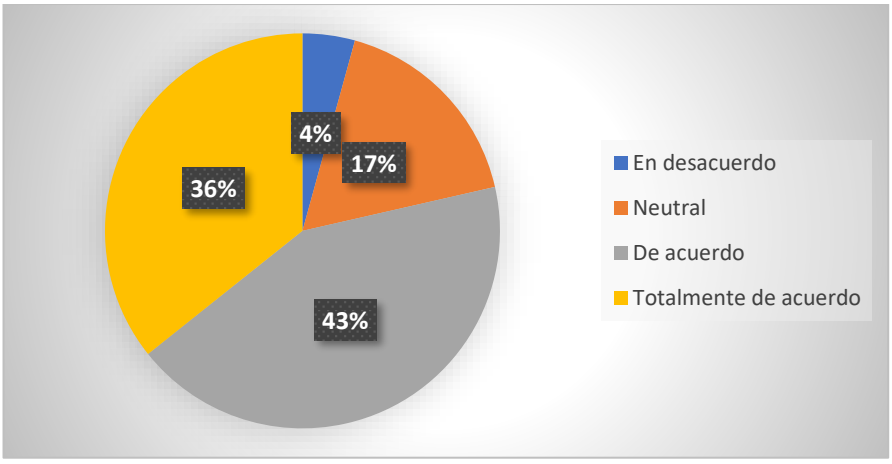


Figura 74: Frecuencia de resultados: pregunta 9, semana 2

En la segunda semana (ver Figura 74), se evidenció una mejora progresiva. El 43% de los usuarios expresó estar de acuerdo y un 36% totalmente de acuerdo, alcanzando un 79% de percepción positiva. El grupo neutral descendió a 17%, y solo un 4% estuvo en desacuerdo. Esta evolución demuestra que el sistema generó mayor confianza a

medida que los usuarios ganaban experiencia. El entorno parece haber resultado más predecible y familiar, lo que reforzó la sensación de seguridad operativa.

Semana 3

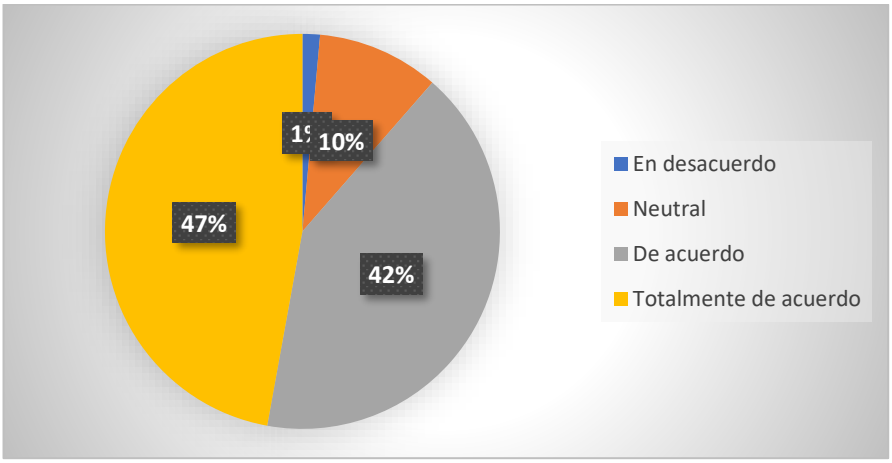


Figura 75: Frecuencia de resultados: pregunta 9, semana 3

La percepción positiva se mantuvo estable. El 42% indicó estar de acuerdo y un 47% totalmente de acuerdo, lo cual representa un 89% de usuarios que se sintieron seguros. Solo un 10% fue neutral, y el 1% restante se mostró en desacuerdo (ver Figura 75). Este fue uno de los momentos más sólidos en cuanto a confianza, reflejando que los usuarios ya dominaban el sistema, conocían sus funciones y sentían que podían operar sin dificultades ni riesgos.

Semana 4

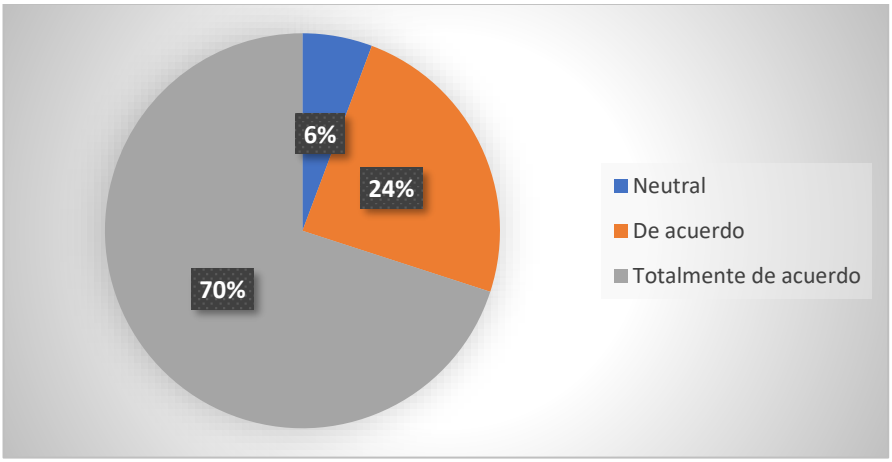


Figura 76: Frecuencia de resultados: pregunta 9, semana 4

En la última semana se alcanzó el punto más alto de seguridad percibida. Un contundente 70% manifestó estar totalmente de acuerdo con sentirse seguros usando el sistema, mientras que un 24% estuvo de acuerdo (ver Figura 76). Solo un 6% se mantuvo neutral y no se registraron respuestas negativas. Esta distribución muestra que el sistema no solo logró ser funcional, sino también generar confianza plena en la experiencia de uso. La evolución a lo largo de las semanas evidencia un progreso claro en la estabilidad emocional y técnica percibida por los usuarios.

Pregunta 10: Necesité aprender muchas cosas antes de poder usar el sistema.

Semana 1

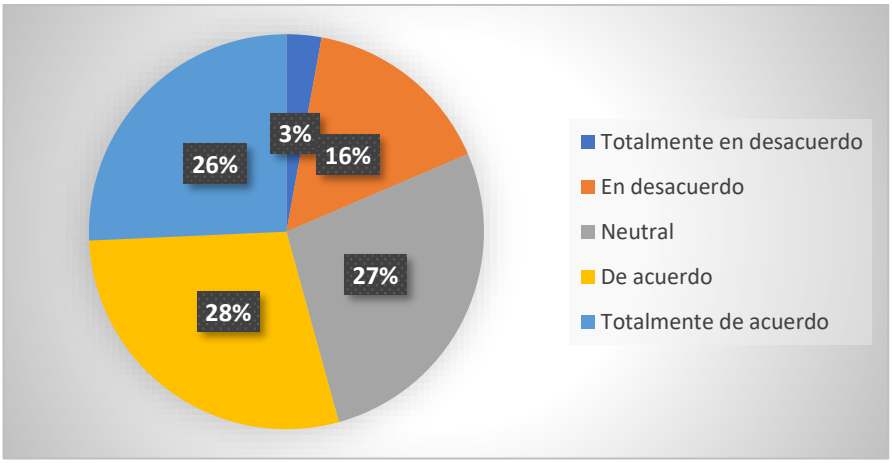


Figura 77: Frecuencia de resultados: pregunta 10, semana 1

En la primera semana (ver Figura 77), los resultados estuvieron muy balanceados. El 28% de los usuarios indicó estar de acuerdo con la afirmación, y un 26% adicional totalmente de acuerdo, lo que refleja que más de la mitad sintió que el sistema requería cierto aprendizaje previo. Por otro lado, un 27% permaneció neutral, mientras que un 16% estuvo en desacuerdo y solo un 3% totalmente en desacuerdo. Estos datos muestran que, al inicio, el sistema no fue percibido como completamente intuitivo y una parte considerable de los usuarios requirió orientación o tiempo de adaptación.

Semana 2

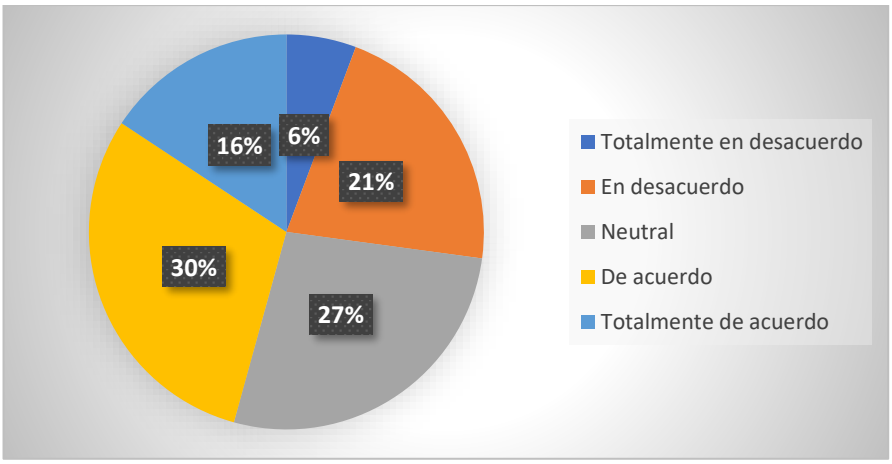


Figura 78: Frecuencia de resultados: pregunta 10, semana 2

En la segunda semana, las percepciones comenzaron a cambiar. Aunque un 30% todavía se mostró de acuerdo y un 16% totalmente de acuerdo con que necesitaban aprender bastante antes de usar el sistema, también aumentó el grupo neutral (27%) y el grupo en desacuerdo (21%). Un 6% indicó estar totalmente en desacuerdo. Estos resultados reflejan que algunos usuarios ya empezaban a familiarizarse mejor con la herramienta, aunque la sensación de complejidad persistía en una parte del grupo (ver Figura 78).

Semana 3

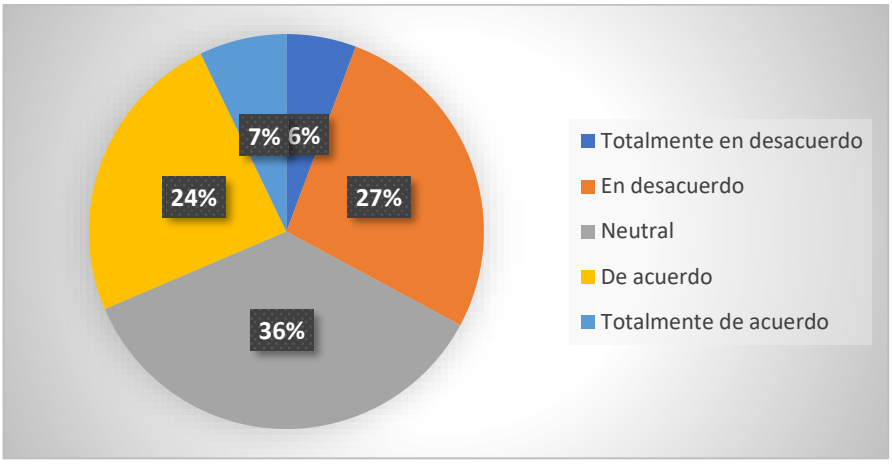


Figura 79: Frecuencia de resultados: pregunta 10, semana 3

Durante la tercera semana, la distribución fue más dispersa. Un 36% de los usuarios se mostró neutral, mientras que el 24% estuvo de acuerdo y un 7% totalmente de acuerdo. En contraste, un 27% estuvo en desacuerdo y un 6% totalmente en desacuerdo. Esta semana muestra una mayor diversidad de experiencias, con algunos usuarios sintiéndose aún poco preparados, mientras que otros ya operaban con autonomía (ver Figura 79). La alta proporción de respuestas neutrales sugiere que la percepción de facilidad seguía en construcción.

Semana 4

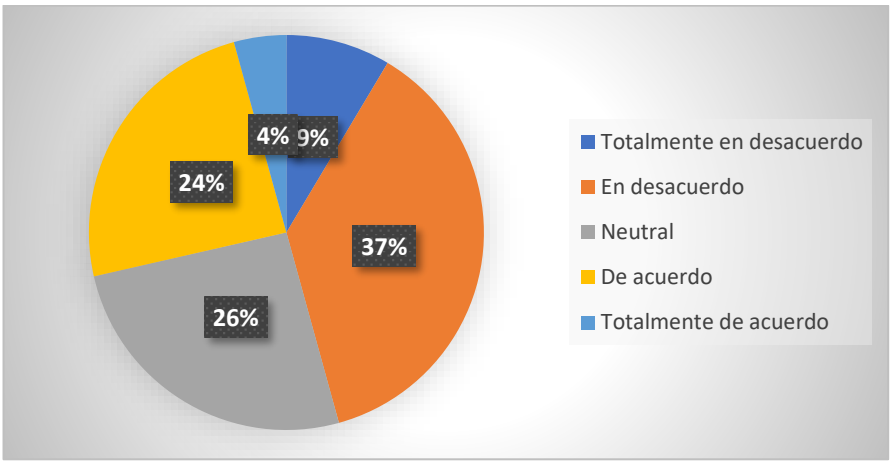


Figura 80: Frecuencia de resultados: pregunta 10, semana 4

En la cuarta y última semana, hubo un leve retroceso en la percepción positiva (ver Figura 80). Un 37% de los usuarios indicó estar en desacuerdo con la afirmación, y solo un 9% estuvo totalmente en desacuerdo, totalizando un 46% de opiniones favorables. Por otra parte, el 26% permaneció neutral, y el 24% expresó estar de acuerdo, con un 4% totalmente de acuerdo. Aunque se mantiene una mayoría que ya no considera necesario aprender mucho para usar el sistema, los datos revelan que aún existe un porcentaje que siente que el uso del sistema requiere un proceso de aprendizaje previo. Esto sugiere oportunidades de mejora en la intuitividad y orientación inicial.

3.7 Índice de Usabilidad (Experiencia de Usuario (UX))

Una puntuación de usabilidad es una métrica que combina diferentes métricas de usabilidad, como la finalización de tareas, el tiempo de tareas, los errores y el índice de satisfacción, en una sola puntuación. Esta puntuación proporciona un porcentaje general que indica el nivel de experiencia del usuario con un producto en función de objetivos de usabilidad predefinidos [95].

Para determinar la intuitividad de una interfaz con realidad virtual significa cuán fácil y natural le resulta a un usuario interactuar con un entorno virtual, sin necesidad de instrucciones complejas o un aprendizaje extenso. En esencia, se trata de medir si la interfaz se "siente" bien y si el usuario puede comprender y utilizar sus funciones de manera instintiva, como si ya supiera cómo hacerlo. Esto implica analizar aspectos como la claridad visual de los elementos, la coherencia de las interacciones, la respuesta a los movimientos del usuario y la capacidad de las acciones virtuales para reflejar las expectativas del mundo real.

a. NASA-TLX

El método NASA TLX (Task Load Index), desarrollado en 1988 por investigadores del Ames Research Center (AMC) y posteriormente revisado (ver Anexo 4: Cuestionario del método NASA TLX (Task Load Index)), es una herramienta subjetiva y multidimensional que permite valorar la carga mental. Este método calcula una puntuación global de la carga de trabajo a partir de la media ponderada de las puntuaciones obtenidas en seis subescalas clave: la exigencia de tipo mental (pensar, decidir, recordar, investigar, etc., y si la tarea es fácil/difícil, simple/compleja, pesada/ligera), la exigencia de tipo físico (empujar, tirar, accionar, etc., y si es fácil/difícil, lenta/rápida, relajada/cansada), la exigencia temporal (presión del tiempo debido al ritmo de las tareas, lento/pausado o rápido/frenético), el esfuerzo (trabajo físico o mental necesario para el rendimiento), y el nivel de frustración (sentimientos de inseguridad, desaliento, irritación, tensión o, por el contrario, seguridad, contento, relajación y satisfacción durante la tarea [78].

Tabla 46: Información técnica y funcional de NASA-TLX para medir carga mental en sistemas interactivos

Aspecto	**Resumen**
Características	Mide carga cognitiva subjetiva en seis dimensiones tras el uso de un sistema
Ventajas	Aporta perspectiva emocional y mental del usuario
Desventajas	Resultados subjetivos; requiere orientación para ponderación correcta
Casos ideales	UI/UX compleja, estudios de usabilidad, validación de interfaces críticas
Compatibilidad	Aplicable en formularios digitales (Google Forms, MATLAB, etc.)
Soporte	Ampliamente aceptado en investigación académica y profesional

La Tabla 46, presenta una descripción del método NASA-TLX, una herramienta ampliamente utilizada para medir la carga cognitiva percibida por los usuarios al interactuar con sistemas informáticos. Su enfoque basado en seis dimensiones proporciona una visión subjetiva del esfuerzo mental y físico requerido, lo que resulta especialmente útil para evaluar la complejidad de interfaces en entornos críticos o de alta demanda cognitiva.

b. Loop11

Loop11 es una herramienta de pruebas de usabilidad remota y en línea. En resumen, Loop 11 te permite crear una encuesta en línea con tareas que los usuarios pueden completar en el sitio web que elijas. Puedes combinar preguntas con tareas en el orden que desees, y para cada tarea puedes establecer un objetivo que el usuario debe alcanzar, por ejemplo, encontrar un jersey azul de mujer talla 48 por menos de 30 £. Las herramientas de usabilidad a distancia son muy valiosas y útiles para el diseño y la administración de tests de usabilidad. Nos pueden servir para simplificar y agilizar los procesos [97].

Tabla 47: Información técnica y funcional de Loop11

Aspecto	Resumen
Características	Herramienta online para pruebas de usabilidad con usuarios reales y tareas específicas
Ventajas	Resultados inmediatos, soporta múltiples dispositivos, útil para UX/UI y pruebas rápidas
Desventajas	No reemplaza pruebas funcionales ni de carga, requiere conexión a internet
Casos ideales	Validación de interfaces, flujos de usuario, pruebas A/B, análisis de formularios o prototipos
Compatibilidad	Compatible con software web, prototipos (Figma, InVision), navegadores modernos
Soporte	Comunidad activa, buena documentación, integración con plataformas como Maze y Hotjar

Como se observa en Tabla 47, describe las capacidades de Loop11, una herramienta enfocada en pruebas de usabilidad con usuarios reales a través de escenarios definidos. Su utilidad radica en la recolección de datos como mapas de calor, clics y tiempos por tarea, lo que facilita la validación temprana de interfaces o prototipos. Es una solución eficaz para equipos de UX, frontend o producto que necesitan tomar decisiones rápidas basadas en evidencia durante ciclos Agile.

c. Mapa comparativo entre Nasa-TLX y Loop 11

La Tabla 48, presenta una comparación entre dos métodos de evaluación de usabilidad: NASA-TLX y Loop11. La tabla muestra distintos aspectos como el tipo de evaluación, enfoque, momento de aplicación, datos generados, utilidad en software, implementación e idoneidad de uso. Se destaca que NASA-TLX se basa en la percepción subjetiva del usuario y evalúa la carga cognitiva posterior al uso del sistema, mientras que Loop11 se enfoca en la interacción objetiva en tiempo real, proporcionando métricas prácticas sobre la usabilidad. Esta comparación permite comprender las diferencias clave entre ambas herramientas según el contexto y propósito de evaluación.

Tabla 48: Mapa comparativo Nasa-TLX vs Loop 11

Aspecto	**NASA-TLX**	**Loop11**
Tipo de evaluación	Subjetiva (percepción del usuario)	Objetiva (interacción del usuario)
Enfoque	Carga cognitiva (mental, física, emocional)	Usabilidad práctica (clics, tiempos, tasa de éxito)
Momento de aplicación	Después de usar el sistema (evaluación post-tarea)	Durante el uso del sistema (en tiempo real)
Datos generados	Opinión ponderada del usuario	Métricas de comportamiento reales
Usabilidad en software	Detecta esfuerzo cognitivo en flujos complejos	Detecta problemas de navegación e interfaz
Implementación	Manual o semiautomática (con formularios)	Plataforma web automatizada
Ideal para	Sistemas críticos o con alta carga mental	Pruebas de interfaces, diseño UX/UI

- **Selección del método de evaluación para el índice de usabilidad**

Después del análisis realizado de los métodos de evaluación de usabilidad Loop11 y NASA-TLX, se procedió a utilizar el método NASA-TLX, ya que este permite medir de manera más adecuada la carga de trabajo percibida por los usuarios mediante cuestionarios estructurados. Esta elección se justificó debido a que Loop11 está orientado principalmente al análisis de interfaces web, lo cual no se ajusta a las características del presente proyecto, que se centra en un entorno de realidad virtual. Por esta razón, NASA-TLX resultó ser la herramienta más apropiada para evaluar la experiencia del usuario en este contexto.

Proceso de aplicación

Cada estudiante, después de utilizar el sistema de simulación del quirófano en realidad virtual, respondió de forma individual el cuestionario NASA-TLX (Anexo 4:

Cuestionario del método NASA TLX (Task Load Index)). Las respuestas fueron recolectadas y tabuladas para obtener un promedio por cada una de las seis dimensiones evaluadas.

Resultados de NASA-TLX

Tabla 49: Promedios obtenidos por los estudiantes en cada dimensión del método NASA-TLX

Dimensión Evaluada	Promedio NASA-TLX
Exigente mental	51.4 %
Exigentes Físicas	52.5 %
Exigentes Temporales	56.1 %
Rendimiento	64.0%
Esfuerzo	48.1%
Frustración	47.4%
Puntuación ponderada total promedio	863.43%

Con base en los resultados obtenidos mediante la aplicación del método NASA-TLX a un total de 70 estudiantes, se logró calcular el promedio de cada una de las dimensiones evaluadas, permitiendo identificar el nivel de carga mental, física y emocional percibido durante el uso del sistema de realidad virtual (ver Tabla 49). En la dimensión Exigente Mental, se obtuvo un promedio de 51.4, lo que indica una carga cognitiva moderada en la interacción con el entorno simulado. En cuanto a las Exigencias Físicas, el promedio alcanzado fue de 52.5, lo cual sugiere que el uso del sistema implicó un esfuerzo físico ligeramente superior al promedio, posiblemente asociado al manejo de los dispositivos de realidad virtual.

Por otro lado, en la dimensión Exigente Temporal, el promedio fue de 56.1, reflejando una percepción de presión moderada en cuanto al tiempo disponible para realizar las tareas. En Rendimiento, se obtuvo un valor de 64.0, lo que evidencia que los estudiantes se sintieron relativamente satisfechos con su desempeño al interactuar con el sistema. La dimensión de Esfuerzo presentó un promedio de 48.1, indicando que los usuarios consideraron que debieron invertir una cantidad moderada de energía para completar sus actividades. Finalmente, la dimensión de Frustración arrojó un promedio de 47.4, lo que muestra que, si bien hubo momentos de dificultad, la sensación general de frustración fue controlada.

En relación con la interpretación de los resultados del método NASA-TLX, se establecen tres niveles de carga mental según la puntuación total obtenida. Un resultado de 500 puntos o menos indica un nivel de carga mental bajo, lo cual sugiere que los usuarios percibieron una carga mínima durante la interacción con el sistema. Si la puntuación se encuentra entre los 501 y los 999 puntos, se considera que existe un nivel de carga mental medio, evidenciando un grado moderado de esfuerzo cognitivo, físico y emocional. Finalmente, una puntuación superior a los 1000 puntos, junto con factores predominantes de carácter intrínseco (aunque su suma no alcance los 100 puntos), corresponde a un nivel de carga mental alto, lo que reflejaría una experiencia significativamente demandante para el usuario. En el caso del presente estudio, la puntuación ponderada total promedio obtenida fue de 863.43, lo cual se ubica dentro del rango de carga mental media. Este resultado sugiere que, en general, los estudiantes experimentaron un nivel moderado de esfuerzo y demanda cognitiva al interactuar con el sistema de simulación de realidad virtual, lo que puede considerarse aceptable para los objetivos académicos planteados.

Estos resultados brindan una visión general sobre la experiencia del usuario durante el uso del sistema de simulación en realidad virtual, permitiendo validar su viabilidad desde una perspectiva de carga de trabajo cognitiva y emocional.

3.8 Metodología Extreme Programming (XP)

Para el desarrollo de este proyecto de tesis, realizado de manera conjunta entre dos estudiantes, se seleccionó la metodología Extreme Programming (XP) debido a su enfoque ágil, su adaptabilidad a cambios frecuentes en los requisitos y su énfasis en la calidad técnica del producto final. Dado que el proyecto consiste en la creación de una aplicación de realidad virtual para simular la organización de la mesa de Mayo en procedimientos quirúrgicos, se requiere una metodología que permita ajustes rápidos durante el desarrollo, iteraciones constantes y validaciones continuas de funcionalidad.

XP ofrece prácticas como la planificación iterativa, la simplicidad en el diseño, el desarrollo incremental y la refactorización continua, las cuales son esenciales para proyectos innovadores donde las necesidades pueden evolucionar conforme avanza el proceso. Además, su filosofía de comunicación constante y revisión colaborativa

resulta adecuada para un trabajo en equipo, permitiendo una mejor coordinación entre los integrantes del proyecto. Estas características convierten a XP en una metodología idónea para alcanzar los objetivos planteados en esta investigación.

La metodología Extreme Programming (XP) se aplicó en el desarrollo de la aplicación de realidad virtual con el objetivo de garantizar la flexibilidad en el diseño y asegurar la calidad del producto final. A continuación, se describen las principales prácticas de XP implementadas durante el proyecto:

Planificación Iterativa (Planning Game)

La planificación de las funcionalidades de la aplicación se realizó de manera iterativa, siguiendo el enfoque del "Planning Game". Al inicio de cada iteración, se definieron historias de usuario basadas en los requisitos del proyecto (como la organización de la mesa de Mayo para procedimientos quirúrgicos). Estas historias representaban tareas pequeñas y manejables, lo que permitió realizar un seguimiento continuo del progreso. Las iteraciones tuvieron una duración de 2 semanas, lo que facilitó la incorporación de cambios rápidos en función de las pruebas y retroalimentación.

Desarrollo Incremental

El desarrollo de la aplicación se realizó en ciclos cortos e incrementales, donde cada entrega al final de la iteración representaba una versión funcional del sistema. Se priorizó la implementación de las funcionalidades más esenciales y fundamentales (como la correcta disposición de los instrumentos en la mesa de Mayo según el tipo de operación), las cuales se fueron agregando y perfeccionando en cada ciclo.

Programación en Pareja (Pair Programming)

Aunque la programación en pareja es una de las prácticas más destacadas de XP, en este proyecto se aplicó de manera parcial debido a las condiciones del equipo de trabajo. Sin embargo, se realizaron sesiones periódicas de revisión de código en las que ambos miembros del equipo trabajaban de manera colaborativa en el análisis de problemas complejos y la mejora del diseño del sistema.

Pruebas Continuas (Test-Driven Development, TDD)

Las pruebas fueron una parte esencial del desarrollo, y se adoptó un enfoque de Desarrollo Guiado por Pruebas (TDD). Antes de escribir cada segmento de código, se definieron pruebas automatizadas para asegurarse de que las funcionalidades de la aplicación, como la correcta disposición de los instrumentos y la interacción con los usuarios, se comportaran según lo esperado. Las pruebas se ejecutaron continuamente, y cualquier error identificado se corrigió de inmediato, evitando la acumulación de defectos en el software.

Refactorización Continua

A medida que avanzaba el proyecto, el código se fue refactorizando para garantizar su simplicidad y eficiencia. La refactorización continua fue aplicada cada vez que se añadían nuevas funcionalidades o se identificaban áreas de mejora, siguiendo la filosofía de XP de evitar la acumulación de deuda técnica. Esto permitió mantener un código limpio y fácil de modificar, lo que facilitó la adaptación a nuevos cambios y requisitos.

Comunicación Constante

A lo largo del desarrollo, se fomentó la comunicación constante entre los dos miembros del equipo. Se realizaron reuniones diarias de corta duración para revisar avances y desafíos, y se utilizó una herramienta de gestión de tareas para mantener el control sobre las historias de usuario y los avances de cada iteración. Esta práctica permitió una respuesta rápida a los cambios y aseguró que ambos miembros del equipo estuvieran alineados durante todo el proceso.

3.8.1 Fases de Extreme Programming (XP)

a. Planificación

En esta fase, el cliente y el equipo definen las funcionalidades prioritarias mediante el análisis de requerimientos, historias de usuario y diagramas de caso de uso, las cuales se priorizan y estiman para ser incluidas en iteraciones cortas. El objetivo es alinear las expectativas y establecer un plan de trabajo flexible.

- ***Análisis de requerimientos***

El análisis de los requerimientos funcionales y no funcionales se fundamenta en la norma ISO 29148:2018, ya que este estándar proporciona directrices actualizadas y completas para la especificación y gestión de requisitos en sistemas y software.

Requerimientos Funcionales

- **RF01:** Pantalla de inicio, el sistema debe mostrar una pantalla de inicio donde tendrá el nombre de la Facultad de Ciencia de la Salud y dos botones: "Iniciar" y "Salir". Al presionar "Salir", se debe cerrar la aplicación. Al presionar "Iniciar", se debe cambiar a la escena del quirófano.

- **RF02**: Creación y personalización del avatar, se desarrollará un avatar el cual debe ser un médico con su respectivo traje al momento de entrar a un quirófano.

- **RF03:** Navegación del avatar, el avatar debe poder caminar, girar y retroceder libremente dentro del quirófano virtual.

- **RF04:** Navegación del avatar, el sistema debe permitir seleccionar entre dos tipos de operación: cesárea y apendicetomía.

- **RF05:** Selección de tiempos quirúrgicos, el sistema debe permitir seleccionar uno de los cuatro tiempos quirúrgicos según el tipo de operación escogido.

- **RF06:** Selección de instrumentos quirúrgicos, el sistema debe permitir seleccionar los instrumentos quirúrgicos correspondientes al tipo de operación y al tiempo quirúrgico.

- **RF07:** Armado de mesa mayo, los instrumentos quirúrgicos seleccionados correctamente se deben colocar automáticamente en la mesa mayo. Si se selecciona un instrumento incorrecto, el sistema debe mostrar un mensaje de advertencia indicando que no corresponde.

- **RF08:** Crear menú de tipo de operación, el sistema debe mostrar un menú en el cual el usuario pueda seleccionar el tipo de operación (cesárea o apendicetomía) y, según la selección, mostrar los tiempos quirúrgicos correspondientes.

- **RF09:** Crear menú de instrumentos quirúrgicos, el sistema debe mostrar un menú con todos los instrumentos quirúrgicos disponibles. Este menú debe permitir al usuario seleccionar los instrumentos necesarios para el tipo de operación y el tiempo quirúrgico previamente seleccionados.

- **RF10:** Creación del escenario quirúrgico, el sistema debe generar un escenario de quirófano realista que incluya todos los elementos propios de una sala quirúrgica, como camilla, lámparas, mesa mayo, mobiliario médico, entre otros. Este escenario debe permitir la navegación del avatar en primera persona.

- **RF11:** Creación del instrumental, para la creación del instrumental se usará solo imágenes en formato SVG, dado que no se despixela al aumentar o disminuir su tamaño.

Requerimientos No Funcionales

- **RNF01:** El sistema debe ser completamente compatible y ejecutable en gafas de realidad virtual Meta Quest 3

- **RNF02:** El quirófano virtual debe tener modelos realistas de acuerdo con el entorno de la vida real, para asegurar una experiencia inmersiva de entrenamiento.

- **RNF03:** Las acciones del usuario, como la selección de operaciones, tiempos e instrumentos deben tener una respuesta del sistema menor a 5 segundo.

- **RNF04:** El sistema debe proporcionar retroalimentación al momento de tomar un instrumento que no pertenezca a un respectivo tiempo.

- **RNF05**: La interfaz debe ser intuitiva, con controles fáciles de aprender y usar, especialmente para usuarios sin experiencia previa en realidad virtual.

- ***Historias de usuario***

Las historias de usuario (user stories) son descripciones breves y simples de una funcionalidad deseada, escritas desde la perspectiva del usuario final [98]. Su propósito principal es:

- Capturar requisitos de forma clara y concisa
- Evitan documentación extensa y técnica, usando lenguaje sencillo
- Priorizar el valor para el usuario
- Ayudan al equipo y al cliente a enfocarse en lo que realmente importa al usuario final, no en detalles técnicos prematuras.
- Promover la comunicación continua
- Permitir flexibilidad y cambios

Tabla 50: Historia de Usuario: Navegar por el menú principal

Historia de usuario	
Numero: 1	Usuario: Estudiante
Nombre Historia: Navegar por el menú principal	
Prioridad: Alta	Riesgo de desarrollo: Alta
Iteración asignada: 1	
Desarrollador: Kevin Saquinga	
Descripción: El estudiante debe poder movilizarse dentro del entrono virtual (delante, atrás, izquierda, derecha) con los mandos de meta quest.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest	

En la Tabla 50, define la necesidad de que el estudiante pueda desplazarse dentro del entorno virtual mediante los controles de Meta Quest. Esta funcionalidad básica de locomoción es crucial para la interacción dentro de la aplicación.

Tabla 51: Historia de Usuario: Navegar por el menú del tutorial

Historia de usuario	
Numero: 2	Usuario: Estudiante
Nombre Historia: Navegar por el menú del tutorial	
Prioridad: Alta	Riesgo de desarrollo: Alta
Iteración asignada: 1	
Desarrollador: Daniel Sánchez	
Descripción: El estudiante debe poder visualizar dentro del entrono virtual los tutoriales para poder entender el funcionamiento de los joysticks, para que una vez que entre al entorno quirúrgico sepa como interactuar con los objetos.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

Esta historia contempla la visualización de tutoriales para que el estudiante entienda el uso de los joysticks antes de ingresar al entorno quirúrgico. En la Tabla 51, se presenta esta historia de usuario, resaltando su importancia en la curva de aprendizaje inicial.

Tabla 52: Historia de Usuario: Navegar por el escenario de simulación quirúrgico

Historia de usuario	
Numero: 3	Usuario: Estudiante
Nombre Historia: Navegar por el escenario de simulación quirúrgico	
Prioridad: Media	Riesgo de desarrollo: Media
Iteración asignada: 1	
Desarrollador: Kevin Saquinga	
Descripción: El estudiante poder navegar dentro del entorno virtual (escenario quirúrgico) y ver cómo está diseñado.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

En la Tabla 52, se describe la necesidad de explorar el entorno quirúrgico simulado, permitiendo al estudiante familiarizarse con el espacio antes de iniciar una práctica.

Como se observa en esta historia el estudiante debe poder escoger el tipo de operación a simular, como cesárea o apendicectomía (ver Tabla 53).

Tabla 53: Historia de Usuario: Selección de operación

Historia de usuario	
Numero: 4	Usuario: Estudiante
Nombre Historia: Selección de operación	
Prioridad: Alta	Riesgo de desarrollo: Media
Iteración asignada: 2	
Desarrollador: Daniel Sánchez	
Descripción: El estudiante debe poder seleccionar el tipo de operación quirúrgica (cesárea, apendicectomía).	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

En la historia de usuario que está en la Tabla 54, permite al usuario seleccionar diferentes tiempos quirúrgicos, facilitando un entrenamiento estructurado.

Tabla 54: Historia de Usuario: Selección de tiempos quirúrgicos

Historia de usuario	
Numero: 5	Usuario: Estudiante
Nombre Historia: Selección de tiempos quirúrgicos	
Prioridad: Alta	Riesgo de desarrollo: Media
Iteración asignada: 2	
Desarrollador: Daniel Sánchez	
Descripción: El estudiante debe poder seleccionar los tiempos quirúrgicos (corte, disección, hemostasia, tracción, separación, aspiración y sutura).	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

Tabla 55: Historia de Usuario: Selección de los instrumentos quirúrgicos

Historia de usuario	
Numero: 6	Usuario: Estudiante
Nombre Historia: Selección de los instrumentos quirúrgicos	
Prioridad: Alta	Riesgo de desarrollo: Alta
Iteración asignada: 2	
Desarrollador: Kevin Saquinga	
Descripción: El estudiante debe poder visualizar y seleccionar los instrumentos quirúrgicos acorde del tiempo quirúrgico y del tipo de operación seleccionado.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

La Tabla 55, contiene la historia que permite a el estudiante identificar y seleccionar los instrumentos quirúrgicos adecuados, según el tipo de operación y tiempo quirúrgico.

Tabla 56: Historia de Usuario: Selección de los instrumentos quirúrgicos

Historia de usuario	
Numero: 7	Usuario: Estudiante
Nombre Historia: Activación de la mesa mayo	
Prioridad: Alta	Riesgo de desarrollo: Alta
Iteración asignada: 3	
Desarrollador: Daniel Sánchez	
Descripción: El estudiante debe activar la mesa quirúrgica (toggle), para comenzar el armado de instrumentos y medir el tiempo total.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

Esta historia permite al estudiante activar la mesa quirúrgica y dar inicio al armado de instrumentos, con cronometraje incluido (ver Tabla 56).

Tabla 57: Historia de Usuario: Selección de los instrumentos quirúrgicos

Historia de usuario	
Numero: 8	Usuario: Estudiante
Nombre Historia: Verificación de errores	
Prioridad: Alta	Riesgo de desarrollo: Alta
Iteración asignada: 3	
Desarrollador: Kevin Saquinga	
Descripción: El estudiante debe recibir mensajes de advertencias si selecciono un instrumento incorrecto, para evitar errores en el armado de la mesa.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

En esta historia se especifica que el sistema debe advertir al estudiante si selecciona un instrumento incorrecto, fomentando la precisión en la simulación (ver Tabla 57).

En la Tabla 58, se detalla esta historia donde se busca mostrar al estudiante el tiempo total utilizado en completar el armado de la mesa, para evaluar su eficiencia.

Tabla 58: Historia de Usuario: Mostrar tiempo total

Historia de usuario	
Numero: 9	Usuario: Estudiante
Nombre Historia: Mostrar tiempo total	
Prioridad: Alta	Riesgo de desarrollo: Alta
Iteración asignada: 3	
Desarrollador: Kevin Saquinga	
Descripción: El estudiante debe visualizar el tiempo total que tardé en completar el armado, para evaluar la eficiencia.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

Tabla 59: Historia de Usuario: Mensaje de finalización

Historia de usuario	
Numero: 10	Usuario: Estudiante
Nombre Historia: Mensaje de finalización	
Prioridad: Media	Riesgo de desarrollo: Media
Iteración asignada: 4	
Desarrollador: Daniel Sánchez	
Descripción: El estudiante debe ver un mensaje al completar todos los tiempos quirúrgicos, para saber que finalicé el proceso correctamente.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

En esta historia, una vez completados todos los tiempos quirúrgicos, el sistema debe mostrar un mensaje de finalización como retroalimentación (ver En la Tabla 58).

Tabla 60: Historia de Usuario: Reiniciar el sistema al volver

Historia de usuario	
Numero: 11	Usuario: Estudiante
Nombre Historia: Reiniciar el sistema al volver	
Prioridad: Media	Riesgo de desarrollo: Media
Iteración asignada: 4	
Desarrollador: Daniel Sánchez	
Descripción: El estudiante al volver al menú principal y comenzar de nuevo, el sistema se reinicie por completo, para tener una nueva simulación limpia.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

En la Tabla 60, se observa la historia que permite reiniciar la simulación al regresar al menú principal, garantizando una nueva práctica desde cero.

Tabla 61: Historia de Usuario: Interacción XR con objetos

Historia de usuario	
Numero: 12	Usuario: Estudiante
Nombre Historia: Interacción XR con objetos	
Prioridad: Alta	Riesgo de desarrollo: Alta
Iteración asignada: 4	
Desarrollador: Daniel Sánchez – Kevin Saquinga	
Descripción: El estudiante al apuntar a un objeto con los punteros de los joysticks debe poder ver su nombre, para identificar los elementos del quirófano.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

Como se observa en la Tabla 61, el estudiante debe poder identificar objetos al apuntarlos con los joysticks, mejorando el aprendizaje visual de los instrumentos.

Tabla 62: Historia de Usuario: Aleatorización de instrumentos

Historia de usuario	
Numero: 13	Usuario: Estudiante
Nombre Historia: Aleatorización de instrumentos	
Prioridad: Alta	Riesgo de desarrollo: Media
Iteración asignada: 4	
Desarrollador: Daniel Sánchez – Kevin Saquinga	
Descripción: El estudiante al momento de armar la mesa mayo los instrumentos quirúrgicos dentro del scroll se ordenen aleatoriamente, para evitar que memorice posiciones y así practicar con mayor realismo.	
Observación: El estudiante debe tener instado el sistema en las gafas Meta quest y tener puesto las gafas.	

Como se observa en la Tabla 62, los instrumentos deben presentarse en orden aleatorio dentro del scroll, para evitar memorización y aumentar el realismo.

- ***Diagramas de Casos de Uso***

Los diagramas de casos de uso son representaciones gráficas que ilustran las interacciones entre los usuarios (actores) y un sistema, mostrando la funcionalidad que el sistema proporciona desde la perspectiva del usuario. Estos diagramas especifican el comportamiento deseado del sistema, pero no indican cómo se implementará dicho comportamiento, constituyendo una herramienta fundamental para la captura y

documentación de requisitos funcionales mediante un lenguaje visual accesible tanto para usuarios técnicos como no técnicos [99].

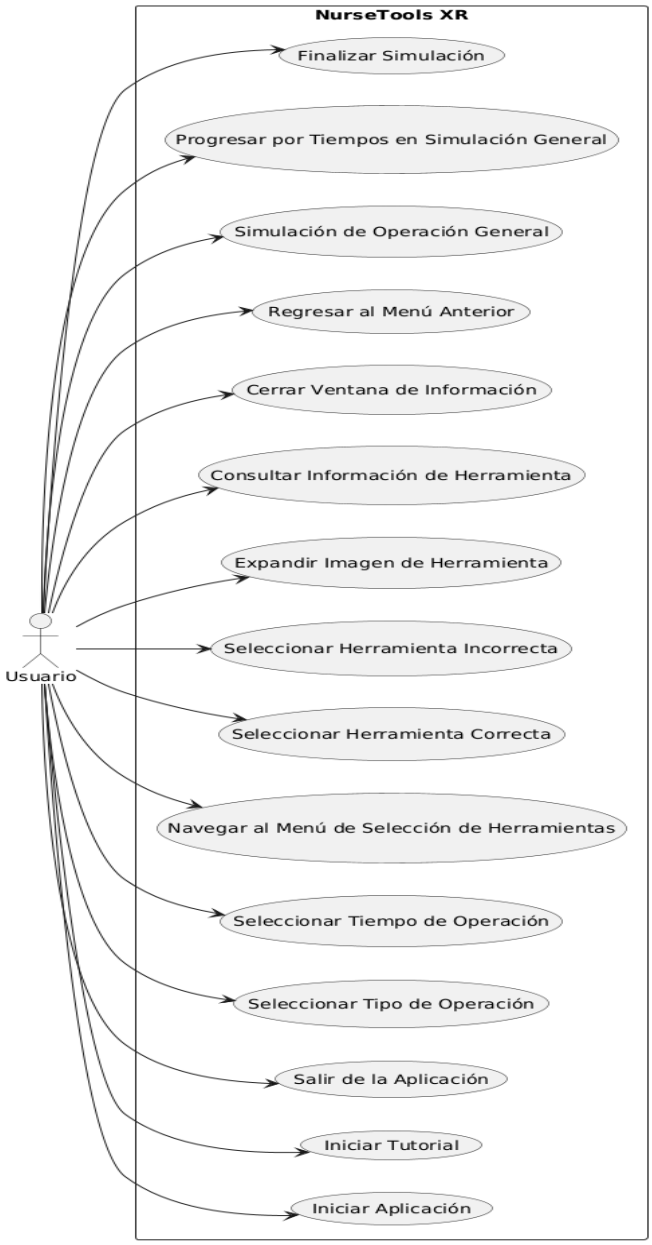


Figura 81: Diagrama de casos de uso

La Figura 81, muestra un diagrama que presenta una vista global de todas las funcionalidades principales que el usuario puede ejecutar dentro de la aplicación de realidad virtual NurseTools XR. Representa al usuario como actor principal que interactúa con los distintos módulos de la simulación, incluyendo el inicio de la aplicación, la selección de tipo y tiempo de operación, la navegación en el menú de herramientas quirúrgicas, la simulación general paso a paso y la consulta de

información sobre los instrumentos. También incluye interacciones relacionadas con el manejo de errores, como la selección incorrecta de herramientas, y acciones auxiliares como expandir imágenes o regresar al menú anterior.

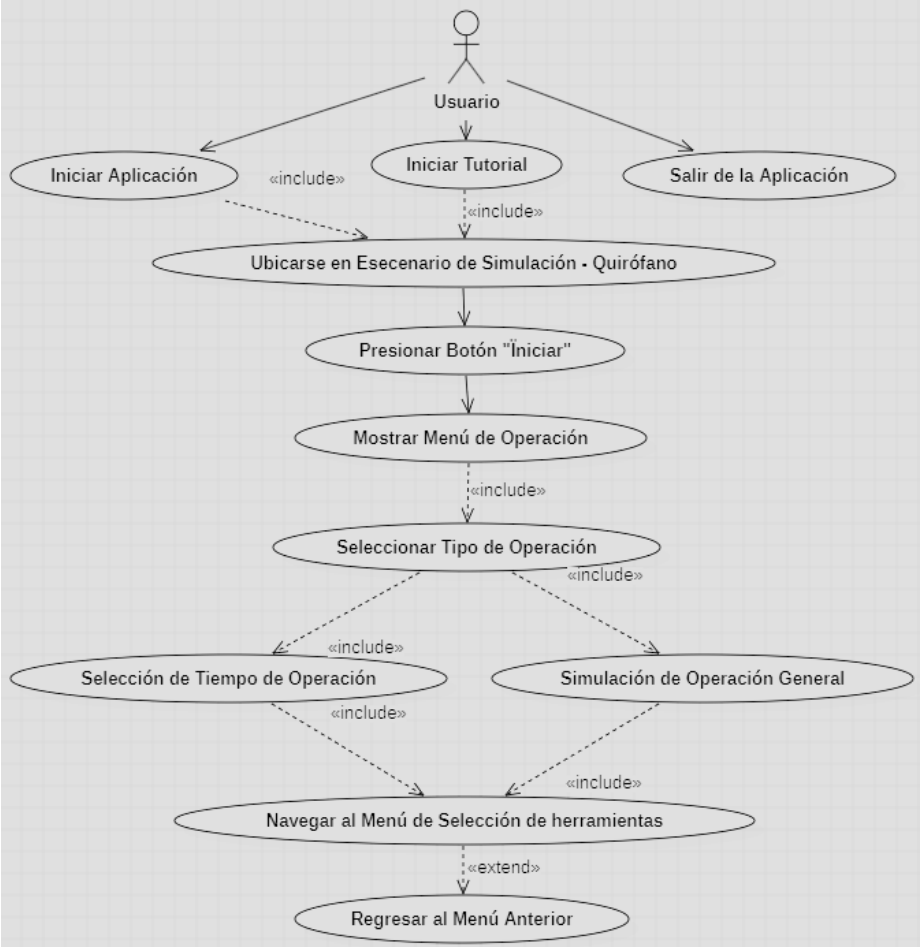


Figura 82: Diagrama de casos de uso: Flujo completo de navegación entre menús

El diagrama de la Figura 82, describe de manera más detallada el flujo de uso desde el momento en que el usuario inicia la aplicación o el tutorial, ubicándose automáticamente en el escenario simulado del quirófano. Desde allí, al presionar el botón "Iniciar" ubicado sobre la mesa de Mayo, se despliega el menú de operación que permite al usuario seleccionar el tipo y tiempo de la cirugía. Luego, el usuario accede al menú de selección de herramientas, desde donde puede regresar al menú anterior si lo desea. Este diagrama refleja el flujo inicial y esencial para configurar correctamente la simulación según los parámetros quirúrgicos seleccionados.

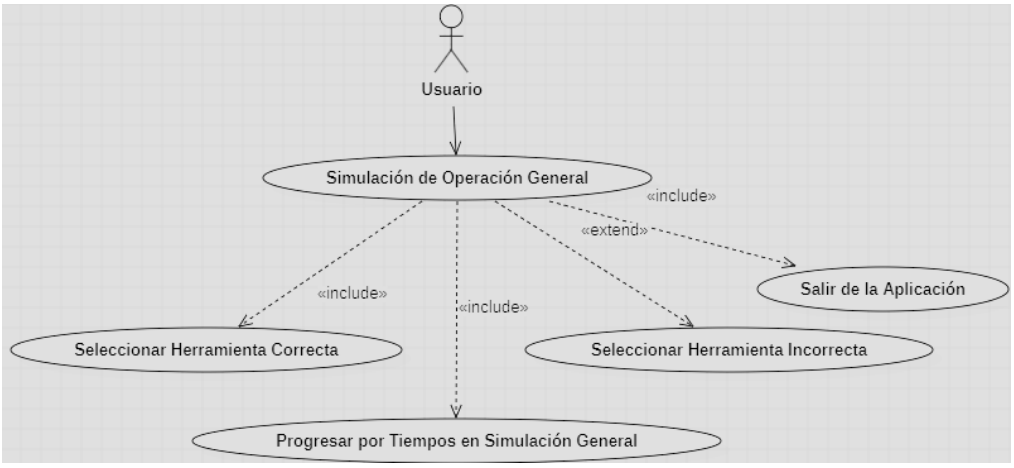


Figura 83: Diagrama de casos de uso: Simulación de Operación General

En la Figura 83, se muestra un diagrama el cual detalla las acciones específicas que el usuario puede llevar a cabo durante la simulación general de una operación. El sistema responde en cada escenario con retroalimentación inmediata. También se contempla la posibilidad de que el usuario salga de la aplicación en cualquier momento, representado como un caso de uso extendido. Este diagrama enfatiza el núcleo funcional de la simulación educativa, centrado en la toma de decisiones y el aprendizaje activo.

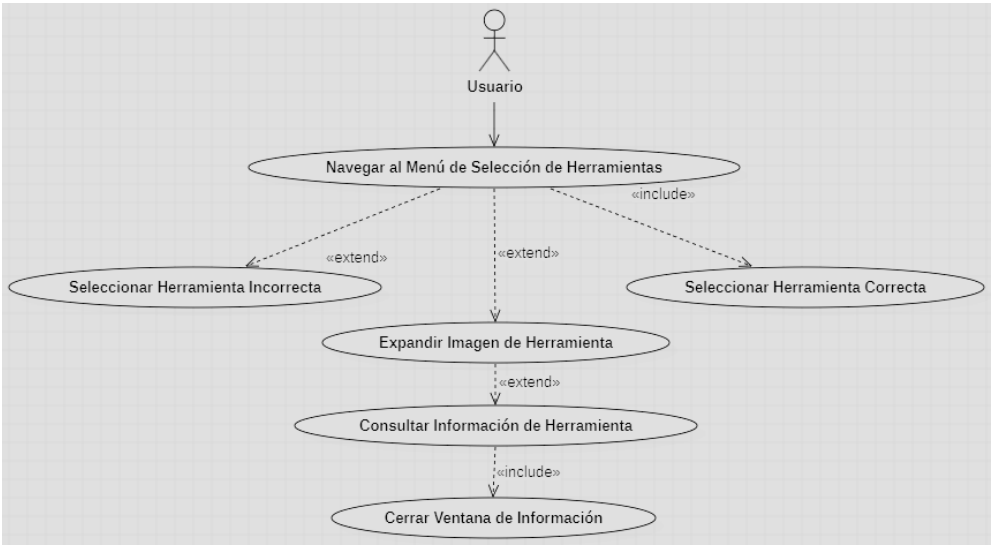


Figura 84: Diagrama de casos de uso: Seleccionar Herramienta Correcta/Incorrecta

El diagrama de casos de uso presentado en la Figura 84, describe la interacción del usuario con el módulo de selección de herramientas dentro de la aplicación de realidad

virtual. En este entorno, el usuario puede navegar por un menú para elegir una herramienta médica, ya sea correcta o incorrectamente. Una vez seleccionada, tiene la opción de expandir la imagen de la herramienta para observarla en detalle y consultar información relevante sobre su uso. Finalmente, puede cerrar la ventana informativa para continuar con la práctica.

b. Diseño

En esta fase, se desarrolla un diseño simple y evolutivo, para mantener la flexibilidad ante cambios futuros. Se emplean técnicas como:

- Tarjetas CRC (Class-Responsibility-Collaboration): Para modelar las interacciones entre objetos y definir responsabilidades clave de cada componente.
- Tarjetas de ingeniería (Engineering Tasks): Donde se desglosan las necesidades técnicas derivadas de las historias de usuario.
- Diagramas de actividades.
- Diagramas de secuencia.

Adicionalmente, se crea un prototipo visual en Figma para validar flujos de usuario y garantizar que el diseño cumpla con los requisitos funcionales. Este enfoque iterativo permite ajustes rápidos basados en feedback continuo, alineando tanto la arquitectura técnica como la experiencia de usuario.

Tarjetas CRC

Las tarjetas CRC es una actividad en grupo de modelado orientado a objetos en la cual el equipo puede manifestar y debatir ideas acerca del diseño de un sistema. Hace especial énfasis en la simplicidad, comunicación y límites de un sistema. Se suele utilizar en las primeras fases del desarrollo de una historia como paso previo a la implementación o escritura de un esquema UML.

Las tarjetas CRC están formadas por tres partes:

- Clase
- Responsabilidades
- Colaboradores

Tabla 63: Tarjeta CRC de MenuPrincipal

MenuPrincipal	
Responsabilidades	**Colaboradores**
Mostrar menú principal Redireccionar a la escena del quirófano Mientas se cargan todos los componentes al cambiar de una escena a otra se activa la carga asíncrona.	CargaAsincrona Escena del quirófano

La clase MenuPrincipal es responsable de mostrar el menú principal y redireccionar al estudiante hacia la escena del quirófano, utilizando carga asincrónica para garantizar una transición fluida entre escenas (ver Tabla 63).

Tabla 64: Tarjeta CRC de CargaAsincrona

CargaAsincrona	
Responsabilidades	**Colaboradores**
Gestionar la carga asincrónica de escenas Mostrar barra de progreso Mostrar mensaje de carga	Escena del quirófano Slider, TMP_Tex

La clase CargaAsincrona se encarga de gestionar la carga asincrónica de escenas, mostrando al usuario una barra de progreso y mensajes informativos durante el proceso (ver Tabla 64).

Tabla 65: Tarjeta CRC de MultiplayerCreationControllerGeneral

MultiplayerCreationControllerGeneral	
Responsabilidades	**Colaboradores**
Capturar selección de operación Mostrar menú de instrumental Reiniciar componentes visuales	ControlTiemposGeneral, AsignarInstrumentoGeneral

Esta clase captura la selección de la operación quirúrgica, muestra el menú de instrumental y reinicia los componentes visuales necesarios para la simulación (ver Tabla 65).

Tabla 66: Tarjeta CRC de ControlTiemposGeneral

ControlTiemposGeneral	
Responsabilidades	**Colaboradores**
Gestionar el flujo por tiempo quirúrgico Contar errores por tiempo Medir el tiempo total de armado	TMP_Text AsignarInstrumentoGeneral

ControlTiemposGeneral es clave para el manejo del flujo por tiempos quirúrgicos, incluyendo la contabilización de errores y la medición del tiempo total de armado (ver Tabla 66).

Tabla 67: Tarjeta CRC de AsignarInstrumentoGeneral

AsignarInstrumentoGeneral	
Responsabilidades	**Colaboradores**
Asignar instrumento al panel según el tiempo Limpiar panel de mesa mayo	ControlTiemposGeneral

Esta clase se ocupa de asignar instrumentos al panel correspondiente según el tiempo quirúrgico y de limpiar la mesa mayo (ver Tabla 67).

Tabla 68: Tarjeta CRC de BotonInstrumentoGeneral

BotonInstrumentoGeneral	
Responsabilidades	**Colaboradores**
Verificar selección válida Enviar instrumento a la mesa Inhabilitar botón luego de seleccionar	ControlTiemposGeneral, AsignarInstrumentoGeneral

BotonInstrumentoGeneral verifica la validez de la selección de instrumentos, envía los instrumentos a la mesa y desactiva el botón correspondiente una vez utilizado (ver Tabla 68).

Tabla 69: Tarjeta CRC de ToggleMesa

ToggleMesa	
Responsabilidades	**Colaboradores**
Activar visualmente la mesa mayo Activar botones de aceptación de instrumentos	MultiplayerCreationControllerGeneral

Esta clase activa visualmente la mesa mayo y habilita los botones para la aceptación de instrumentos, permitiendo el inicio de la simulación de armado (ver Tabla 69).

Tabla 70: Tarjeta CRC de MensajeErrores

MensajeErrores	
Responsabilidades	**Colaboradores**
Mostrar resumen de errores por tiempo Mostrar tiempo total y operación	ControlTiemposGeneral

La clase MensajeErrores tiene como responsabilidad mostrar un resumen de errores cometidos por tiempo quirúrgico y visualizar el tiempo total y tipo de operación realizada (ver Tabla 70).

Tabla 71: Tarjeta CRC de MenuOperacion

MenuOperacion	
Responsabilidades	**Colaboradores**
Capturar tipo de operación y tiempo inicial Activar menú siguiente	MultiplayerCreationControllerGeneral

En la clase MenuOperacion tiene como responsabilidad la captura del tiempo de operación y tiempo quirúrgico, también la activación de los menú donde se encuentra los instrumentos quirúrgicos (ver Tabla 71).

Tabla 72: Tarjeta CRC de PanelInstrumentos

PanelInstrumentos	
Responsabilidades	**Colaboradores**
Alojar botones de instrumentos Desordenar instrumentos al inicio	MultiplayerCreationControllerGeneral

PanelInstrumentos aloja los botones correspondientes a los instrumentos quirúrgicos y realiza su aleatorización al iniciar la simulación (ver Tabla 72).

Tabla 73: Tarjeta CRC de ControlMesaMayo

ControlMesaMayo	
Responsabilidades	**Colaboradores**
Gestionar la visualización/ocultación de la mesa de mayo y sus componentes Controlar el estado de los botones de selección de instrumentos Limpiar y reiniciar la mesa	ControlTiemposGeneral botonesCheckInstrumentos

Esta clase gestiona la visualización y limpieza de la mesa mayo, controlando el estado de los botones de selección de instrumentos (ver Tabla 73).

Tabla 74: Tarjeta CRC de ControladorExpandir

ControladorExpandir	
Responsabilidades	**Colaboradores**
Visualización de imágenes en modo expandido Visualización de información adicional	TMP_Text Image panelInformacion

ControladorExpandir permite visualizar imágenes en modo ampliado y mostrar información adicional asociada a los elementos quirúrgicos (ver Tabla 74).

Tarjetas de ingeniería (Componentes)

Las tarjetas de ingeniería (también llamadas Technical Tasks o Tech Stories) son elementos de planificación que descomponen requerimientos técnicos necesarios para implementar una funcionalidad. A diferencia de las historias de usuario (centradas en el valor para el cliente), estas tarjetas se enfocan en cómo se construirá algo desde el lado técnico.

Tabla 75: Tarjeta de Ingeniería de MenuPrincipal

Componente: MenuPrincipal	
Descripción:	Interfaz inicial del sistema con botones para ingresar a la escena de instrumentación, iniciar simulación, tutorial o salir.

El componente MenuPrincipal representa la interfaz inicial de la aplicación, permitiendo al usuario seleccionar entre diferentes opciones como iniciar la simulación, ingresar al tutorial o salir del sistema (ver Tabla 75).

Tabla 76: Tarjeta de Ingeniería de CargaAsincrona

Componente: CargaAsincrona	
Descripción:	Controla la carga asincrónica de escenas, desplegando barra de progreso y texto informativo

CargaAsincrona es responsable de gestionar la carga de escenas de manera asincrónica, mostrando una barra de progreso y mensajes informativos durante la transición (ver Tabla 76).

Tabla 77: Tarjeta de Ingeniería de MenuOperacion

Componente: MenuOperacion	
Descripción:	Pantalla de selección de tipo de operación y tiempo quirúrgico inicial.

Este componente permite al usuario seleccionar el tipo de operación quirúrgica y definir el tiempo quirúrgico inicial antes de iniciar la simulación (ver Tabla 77).

Tabla 78: Tarjeta de Ingeniería de MenuInstrumentalGeneral

Componente: MenuInstrumentalGeneral	
Descripción:	Contenedor de selección de instrumentos por tiempo. Muestra operación, tiempo, errores y toggle de activación.

MenuInstrumentalGeneral funciona como contenedor de selección de instrumentos, mostrando información clave como el tiempo quirúrgico, número de errores y la opción de activar la mesa (ver Tabla 78).

Tabla 79: Tarjeta de Ingeniería de ControlTiemposGeneral

Componente: ControlTiemposGeneral	
Descripción:	Clase que administra la lógica del avance de tiempos quirúrgicos, errores y tiempo total.

Este componente administra el flujo lógico de los tiempos quirúrgicos, registrando errores y contabilizando el tiempo total de armado (ver Tabla 79).

Tabla 80: Tarjeta de Ingeniería de AsignarInstrumentoGeneral

Componente: AsignarInstrumentoGeneral	
Descripción:	Clase que gestiona el almacenamiento y posicionamiento de los instrumentos seleccionados.

AsignarInstrumentoGeneral se encarga de gestionar el almacenamiento y posicionamiento adecuado de los instrumentos seleccionados durante la práctica (ver Tabla 80).

Tabla 81: Tarjeta de Ingeniería de ToggleMesa

Componente: ToggleMesa	
Descripción:	Componente UI que activa la visibilidad del canvas de la mesa y errores.

El componente ToggleMesa permite activar visualmente el canvas de la mesa mayo, facilitando la visualización de errores y el acceso a los instrumentos (ver Tabla 81).

Tabla 82: Tarjeta de Ingeniería de PanelMesaMayo

Componente: PanelMesaMayo	
Descripción:	Canvas que contiene los instrumentos ya seleccionados agrupados por tiempo quirúrgico.

Este panel contiene los instrumentos ya seleccionados, organizados según el tiempo quirúrgico correspondiente (ver Tabla 82).

Tabla 83: Tarjeta de Ingeniería de PanelErrores

Componente: PanelErrores	
Descripción:	Canvas que muestra el total de errores por tiempo quirúrgico y el tiempo total.

PanelErrores presenta un resumen del total de errores cometidos durante cada etapa quirúrgica, así como el tiempo total registrado (ver Tabla 83).

Tabla 84: Tarjeta de Ingeniería de BotonInstrumentoGeneral

Componente: BotonInstrumentoGeneral	
Descripción:	Script en cada botón de instrumento que verifica validez, asigna el Sprite y actualiza errores.

Cada botón de instrumento cuenta con un script que, valida la selección, asigna su representación visual y actualiza el conteo de errores (ver Tabla 84).

Tabla 85:Tarjeta de Ingeniería de PanelInstrumentos

Componente: PanelInstrumentos	
Descripción:	Canvas que muestra los instrumentos quirúrgicos, también te permitirá seleccionar para ir armando la mesa mayo.

Este canvas presenta todos los instrumentos quirúrgicos disponibles, permitiendo al usuario seleccionar aquellos necesarios para armar la mesa (ver Tabla 85).

Tabla 86: Tarjeta de Ingeniería de ControlMesaMayo

Componente: ControlMesaMayo	
Descripción:	Script que gestiona la visualización/ocultación de la mesa de mayo y sus componentes.

ControlMesaMayo gestiona la visibilidad de la mesa mayo y sus elementos, además de permitir su reinicio para nuevas simulaciones (ver Tabla 86).

Tabla 87: Tarjeta de Ingeniería de ControladorExpandir

Componente: ControladorExpandir	
Descripción:	Script en cada botón de los instrumentos que gestiona la visualización de imágenes en modo expandido y de la visualización de información adicional

Este componente permite al usuario ampliar visualmente una imagen de un instrumento y acceder a información adicional sobre su uso (ver Tabla 87).

Tabla 88: Tarjeta de Ingeniería de Aleatorización del ScrollView

Componente: Aleatorización del ScrollView	
Descripción:	Las imágenes del instrumental quirúrgico se deben aleatorizar al abrir el menú de instrumentos.

Este componente tiene la tarea de aleatorizar la posición de los instrumentos dentro del menú, impidiendo que el estudiante memorice ubicaciones y fomentando el aprendizaje realista (ver Tabla 88).

Diseño de prototipado en Figma

Figma es una herramienta de diseño colaborativo basada en la nube, utilizada para crear interfaces de usuario, prototipos y sistemas de diseño. Permite trabajar en tiempo real con equipos multidisciplinarios, facilitando la creación de wireframes, flujos de interacción y diseños visuales. En el desarrollo del sistema, se empleó Figma para diseñar los componentes de la interfaz, como los paneles de la mesa de mayo y los elementos expandibles, asegurando una experiencia de usuario intuitiva y coherente antes de su implementación en Unity. Esta herramienta ayudó a validar el diseño de manera iterativa, incorporando feedback temprano y optimizando el flujo de trabajo entre diseñadores y desarrolladores.



Figura 85: Menú Principal del sistema NurseTools

Según la Figura 85, se puede observar el menú principal que se presentará en el sistema de realidad virtual. Este menú permitirá al usuario acceder a las distintas funcionalidades del entorno, como tutoriales, salir y el inicio de la simulación. Su diseño está pensado para ofrecer una navegación intuitiva y clara, facilitando la experiencia del usuario desde el primer contacto con el sistema.



Figura 86: Tutorial de cómo usar el movimiento de los joysticks

Como se observa en la Figura 86, se muestra cómo quedará el canvas del tutorial para el control de movimientos dentro del entorno virtual. Este tutorial guiará al usuario sobre cómo desplazarse correctamente por el escenario utilizando los controles del

sistema. Se ha diseñado con instrucciones visuales claras y accesibles, que permitan al usuario familiarizarse rápidamente con la mecánica de navegación.



Figura 87: Tutorial de cómo activar los punteros

En la Figura 87, se presenta el canvas del tutorial que explica cómo activar los punteros dentro del escenario del quirófano. Este tutorial es esencial para enseñar al usuario cómo utilizar los punteros para interactuar con los objetos del entorno. Su implementación busca facilitar la manipulación precisa de instrumentos y otros elementos interactivos dentro de la simulación.



Figura 88: Tutorial de como interactuar dentro del escenario del quirófano

La Figura 88, se muestra el canvas del tutorial que enseña cómo interactuar con los objetos seleccionados en el escenario. Su propósito es garantizar una interacción fluida y realista con los elementos del entorno quirúrgico.

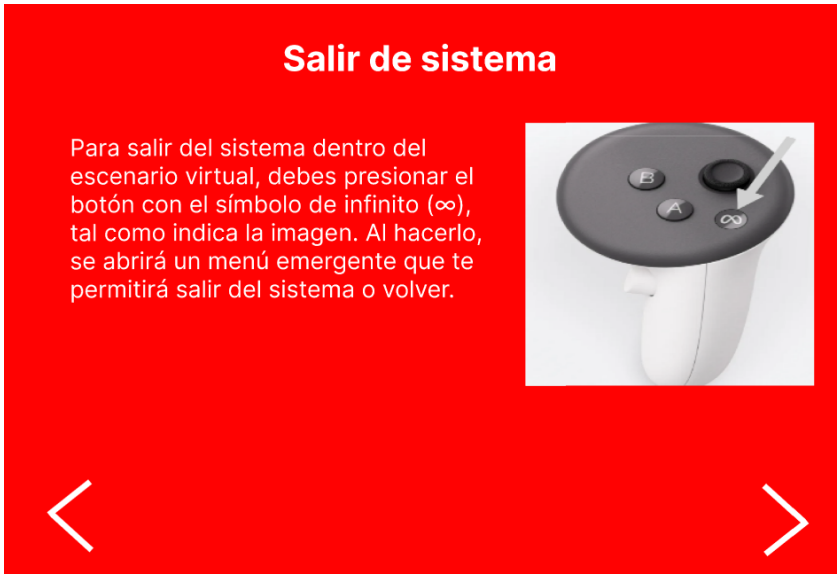


Figura 89: Tutorial de cómo salir del sistema

Según la Figura 89, se presenta el canvas del tutorial que indica cómo salir correctamente del sistema de realidad virtual. Este tutorial proporciona los pasos necesarios para cerrar la sesión de manera.

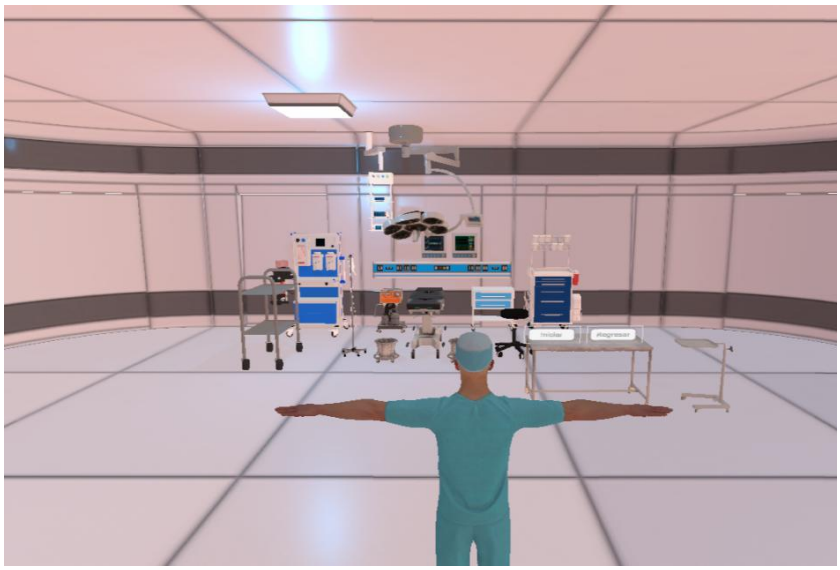


Figura 90: Escenario del Quirófano

En la Figura 90, se visualiza cómo quedará el escenario del quirófano dentro del sistema de realidad virtual. Este espacio ha sido diseñado con un alto nivel de detalle para simular un entorno quirúrgico realista y funcional.



Figura 91: Menú Nurse Tools que te permite seleccionar el tipo de operación y tiempo

Como se aprecia en la Figura 91, se muestra el canvas del menú que permitirá seleccionar el tipo de operación y el tiempo quirúrgico. Este menú proporciona al usuario diversas opciones para configurar el procedimiento que desea simular. Facilita la personalización de la experiencia según los objetivos de aprendizaje definidos.

Se observa en la Figura 92, el canvas del menú de selección de los instrumentos quirúrgicos.



Figura 92: Menú del instrumental quirúrgico

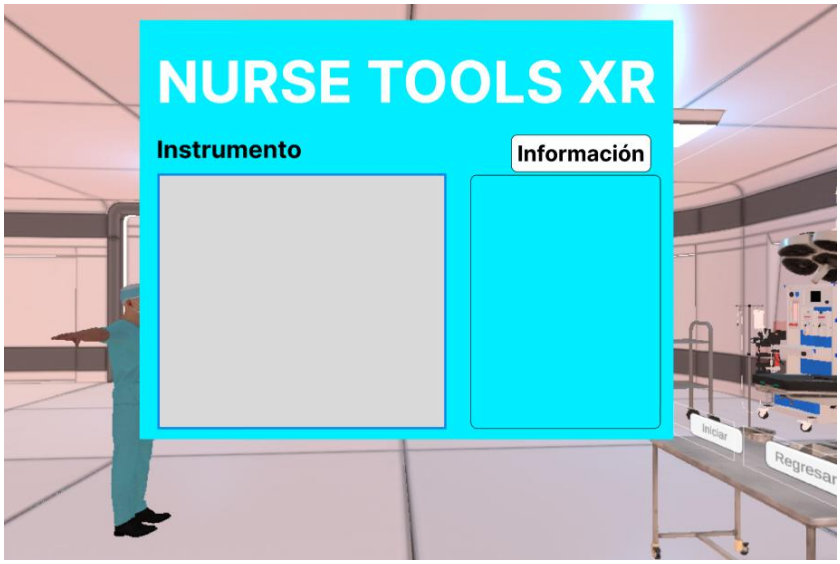


Figura 93: Canvas para ver información extra de la imagen

Según la Figura 93, se muestra el canvas que permite ver la imagen expandida y obtener información adicional del instrumento seleccionado. Esta función es útil para conocer más detalles sobre el uso y las características de cada herramienta médica. Contribuye al aprendizaje teórico y práctico del estudiante dentro del entorno simulado.

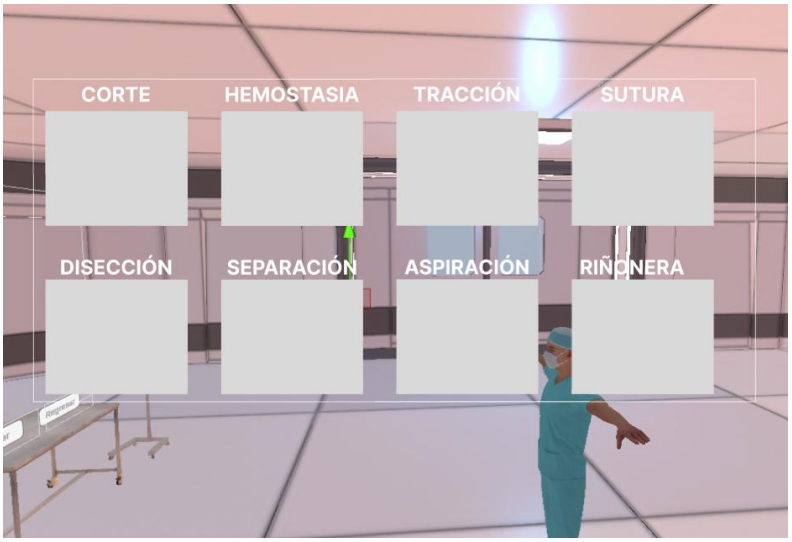


Figura 94: Canva de la mesa Mayo

Como se muestra en la Figura 94, se presenta el canvas correspondiente a la mesa mayo, la cual se irá llenando conforme se seleccionen los instrumentos en el menú. Este componente refleja de manera visual y organizada la disposición de los instrumentos quirúrgicos disponibles para el procedimiento.

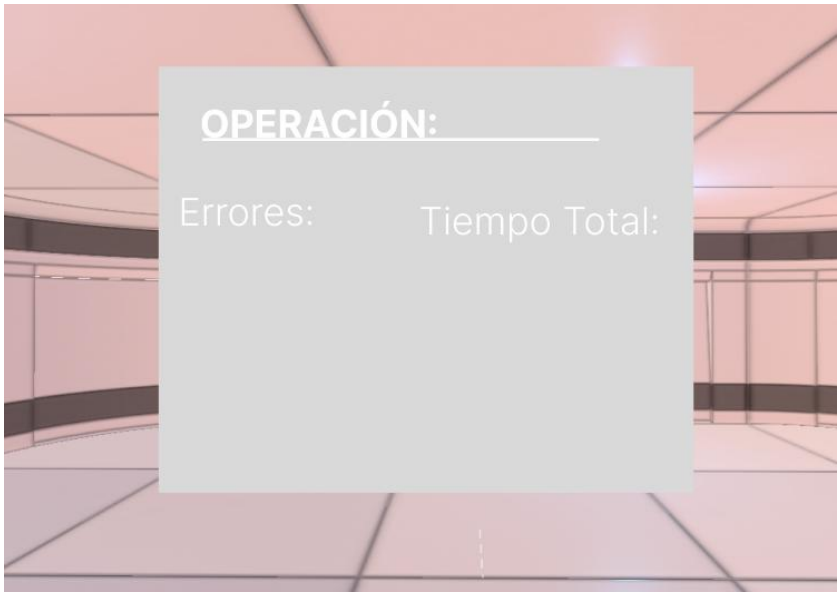


Figura 95: Canva para el control de errores y tiempo total

La Figura 95, exhibe el canvas destinado al control de errores y el tiempo total que el usuario ha demorado en completar la actividad. Este apartado permite evaluar el desempeño del estudiante durante la simulación, proporcionando retroalimentación

inmediata. Los datos recogidos pueden ser utilizados para reforzar el aprendizaje y mejorar la toma de decisiones clínicas.

- ***Diagramas de Actividades***

Los diagramas de actividades son representaciones del flujo de trabajo que modelan el comportamiento dinámico de un sistema, mostrando la secuencia de actividades, decisiones y concurrencias en los procesos. Estos diagramas muestran el flujo de control general y permiten visualizar el comportamiento de un sistema con énfasis en la secuencia y condiciones de coordinación de comportamientos de bajo nivel, siendo particularmente útiles para modelar procesos de negocio, flujos de trabajo y algoritmos complejos en el desarrollo de software [100].

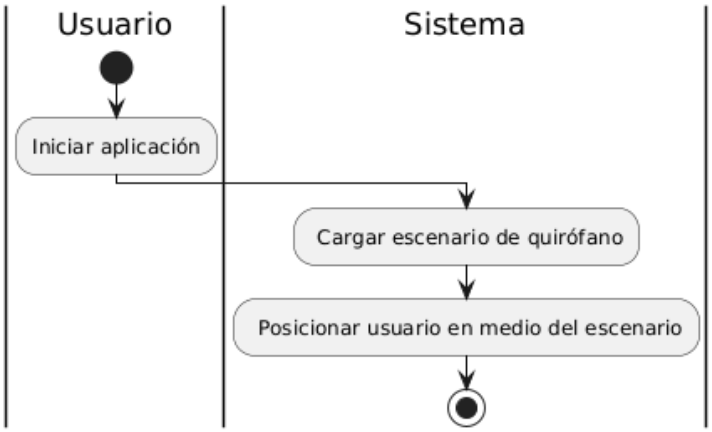


Figura 96: Diagrama de actividad: Inicio y Posicionamiento en el Escenario

Como se muestra en la Figura 96, la aplicación inicia cargando un escenario de quirófano virtual en 3D, donde el sistema posiciona automáticamente al usuario en el centro del entorno para iniciar la interacción.

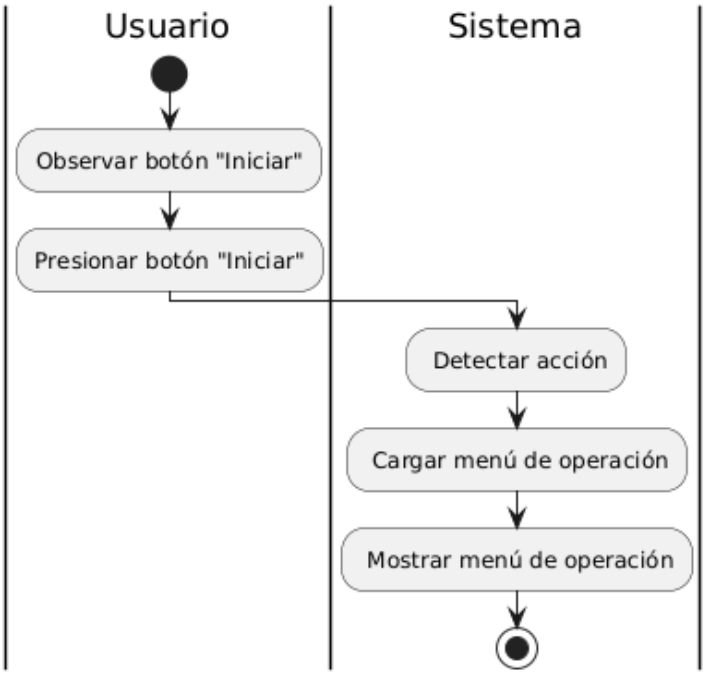


Figura 97: Diagrama de actividad: Presionar Botón "Iniciar" y Abrir Menú de Operación

La Figura 97, ilustra el flujo de acciones donde, al presionar el botón "Iniciar", el sistema despliega un menú interactivo con las opciones de procedimientos quirúrgicos disponibles para el usuario.

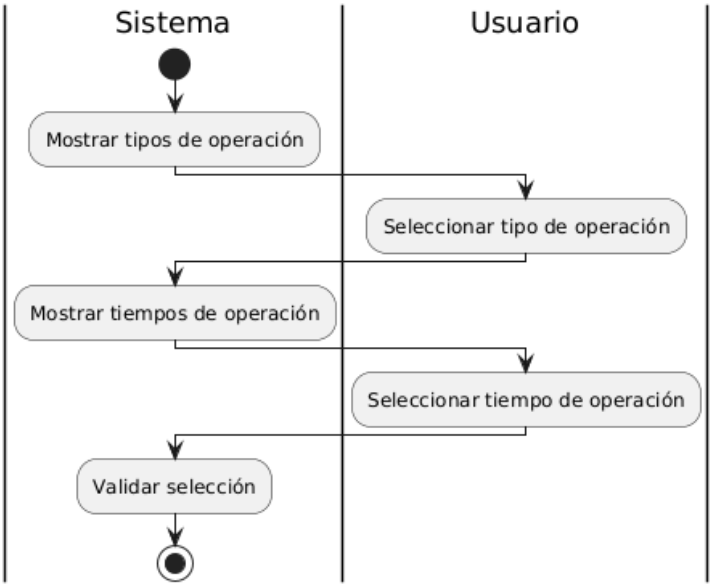


Figura 98: Diagrama de actividad: Selección de Tipo y Tiempo de Operación

Como se detalla en la Figura 98, el usuario define los parámetros de la simulación seleccionando el tipo y el tiempo de la operación, mientras el sistema valida estas opciones para asegurar coherencia en el entrenamiento virtual.

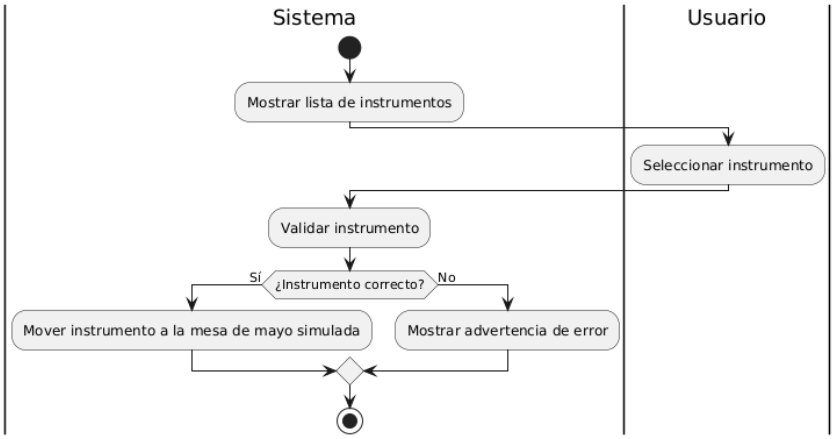


Figura 99: Diagrama de actividad: Selección de Instrumentos

El diagrama de la Figura 99, describe cómo, al seleccionar un instrumento, el sistema verifica su correspondencia con la operación en curso. Si es válido, lo ubica en la mesa de mayo; de lo contrario, muestra una advertencia de error.

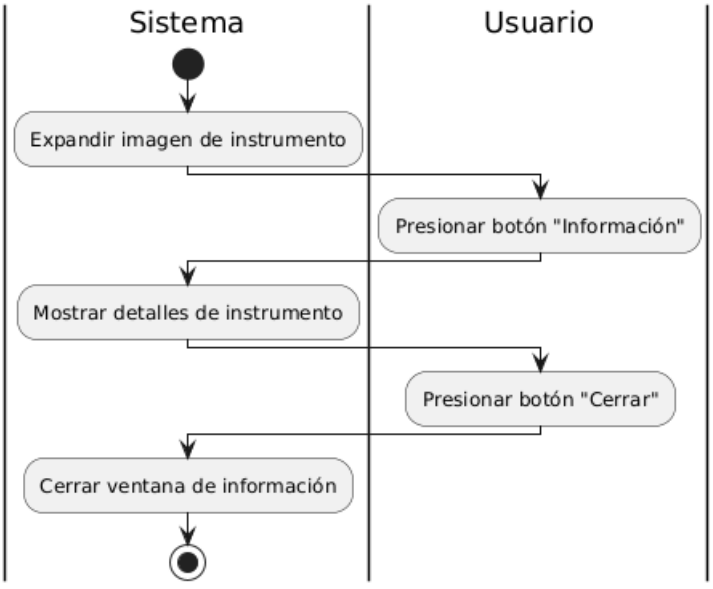


Figura 100: Diagrama de actividad: Consulta de Información de Instrumentos

En la Figura 100, se observa el proceso de exploración de instrumentos médicos, donde el usuario puede ampliar su visualización y acceder a una ventana emergente con detalles técnicos, la cual puede cerrarse cuando ya no sea necesaria.

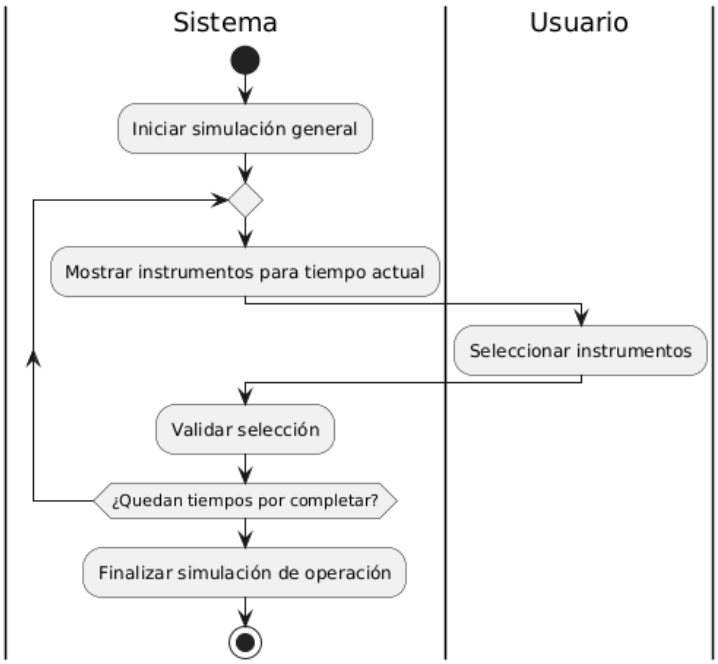


Figura 101: Diagrama de actividad: Simulación Completa de la Operación

El diagrama de actividad de la Figura 101, representa el flujo de la simulación general dentro de la aplicación de realidad virtual. El proceso inicia con la presentación de los instrumentos disponibles según el tiempo actual de la simulación, permitiendo al usuario seleccionar los adecuados. Luego, el sistema valida la selección y verifica si aún hay tiempos pendientes por completar. Este ciclo se repite hasta finalizar la operación.

- ***Diagramas de Secuencia***

Los diagramas de secuencia son representaciones temporales que muestran la interacción entre objetos mediante un intercambio ordenado de mensajes en el tiempo. Estos diagramas ilustran cómo los objetos colaboran en un escenario particular de un caso de uso, mostrando la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la funcionalidad descrita por el escenario, resultandos fundamentales para diseñar y comprender las interacciones dinámicas entre componentes de un sistema orientado a objetos[101].

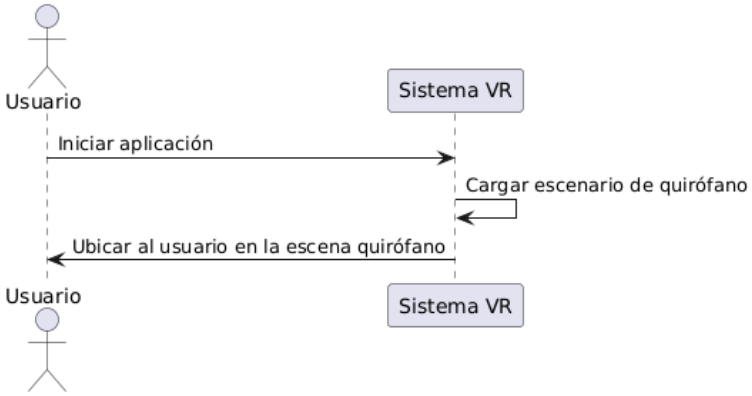


Figura 102: Diagrama de secuencia: Inicio de la Aplicación y Posicionamiento en el Escenario

En la Figura 102, el diagrama de secuencia ilustra el proceso inicial de interacción entre el usuario y el sistema de realidad virtual (VR). El flujo comienza cuando el usuario inicia la aplicación, lo que desencadena la acción del sistema VR para cargar el escenario correspondiente al quirófano. Una vez completada esta acción, el sistema ubica al usuario dentro de dicha escena, permitiéndole comenzar la experiencia inmersiva. Este diagrama refleja la sincronización de mensajes necesarios para establecer el entorno virtual de manera automática y fluida.

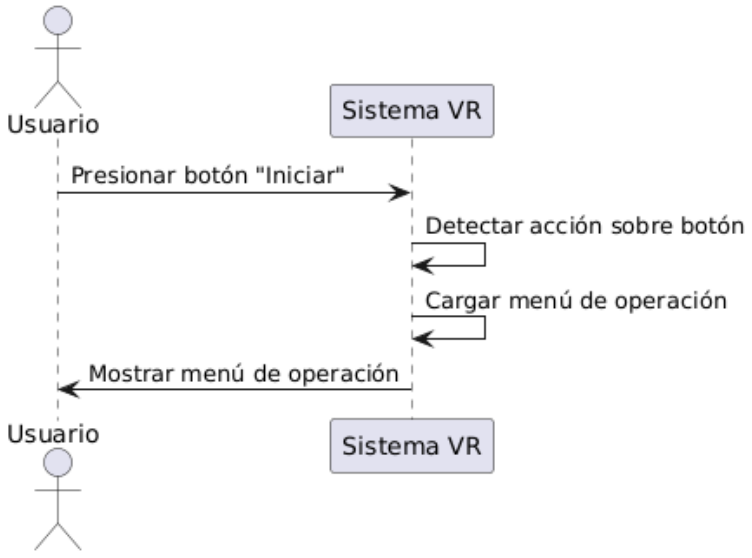


Figura 103: Diagrama de secuencia: Presionar Botón "Iniciar"

En la Figura 103, se muestra el diagrama de secuencia correspondiente al inicio de la simulación en la aplicación de realidad virtual. El usuario interactúa con el sistema al presionar el botón "Iniciar", lo cual desencadena una serie de eventos dentro del

sistema VR: se detecta la acción del usuario, se carga el menú de operación y finalmente se presenta visualmente al usuario. Este proceso refleja la lógica básica de interacción para dar inicio a la experiencia inmersiva.

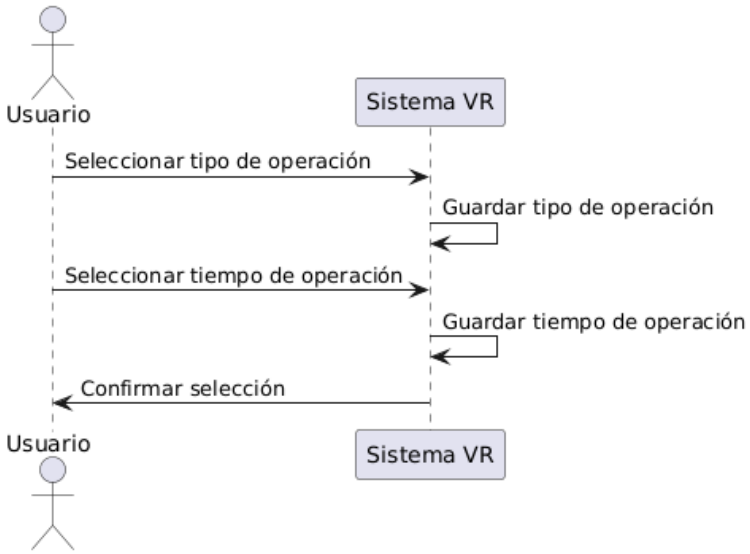


Figura 104: Diagrama de secuencia: Seleccionar Tipo y Tiempo de Operación

La Figura 104, ilustra el flujo de eventos que tiene lugar cuando el usuario selecciona los parámetros específicos de la operación que se desea simular. El usuario primero elige el tipo de operación y luego el tiempo estimado de duración. El sistema VR guarda ambos datos y, tras la confirmación por parte del usuario, finaliza esta etapa del proceso. Este diagrama resalta la configuración personalizada de la experiencia, adaptando la simulación a las preferencias del usuario.

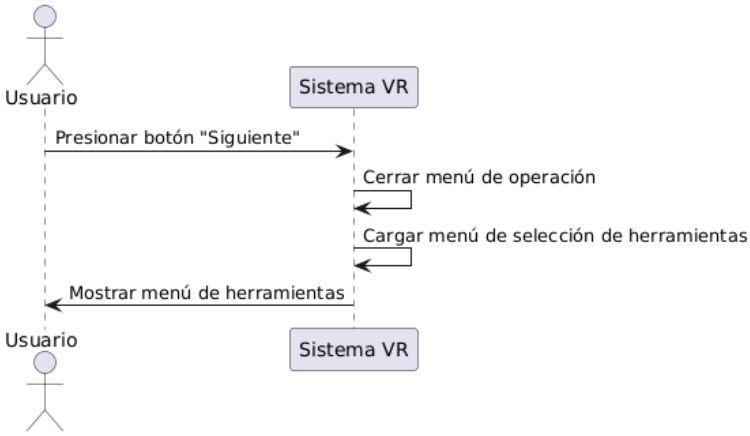


Figura 105: Diagrama de secuencia: Navegar al Menú de Selección de Herramientas

En la Figura 105, se presenta el diagrama de secuencia que representa la transición hacia la selección de instrumentos quirúrgicos. Al presionar el botón "Siguiente", el sistema VR responde cerrando el menú de operación previamente utilizado y cargando el nuevo menú de herramientas. Finalmente, el menú de selección de herramientas se muestra al usuario, permitiendo la interacción con los instrumentos necesarios para la simulación. Esta etapa es clave para preparar al usuario antes de iniciar la práctica quirúrgica.

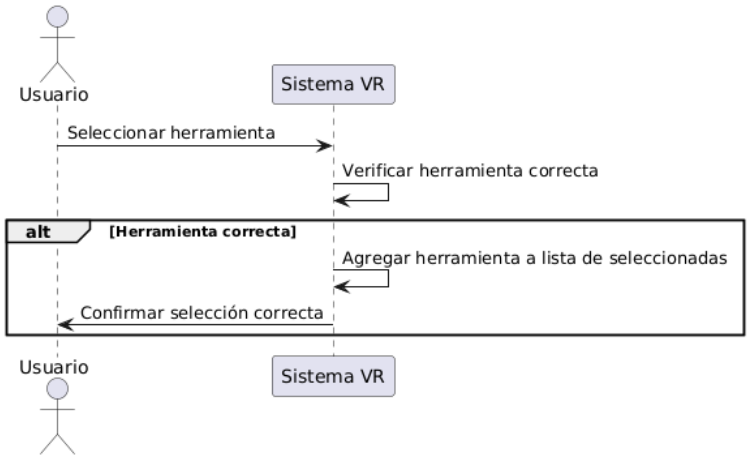


Figura 106: Diagrama de secuencia: Selección Correcta de Herramientas

En la Figura 106, se presenta el diagrama de secuencia correspondiente al proceso de selección correcta de una herramienta dentro del sistema de realidad virtual (VR). El usuario interactúa con la interfaz para seleccionar una herramienta, acción que es evaluada por el sistema. Si la herramienta seleccionada es la adecuada para la actividad en curso, el sistema la agrega a una lista de herramientas seleccionadas y confirma al usuario que la elección fue correcta. Este flujo garantiza una retroalimentación inmediata y adecuada durante el entrenamiento.

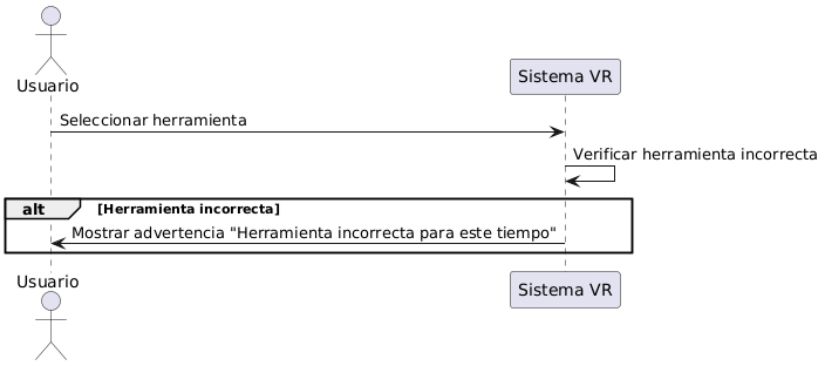


Figura 107: Diagrama de secuencia: Selección Incorrecta de Herramientas

La Figura 107, muestra el diagrama de secuencia asociado al caso en que el usuario selecciona una herramienta incorrecta. El sistema VR verifica la validez de la selección y, al detectar un error, responde mostrando un mensaje de advertencia que indica que la herramienta no es apropiada para ese momento del procedimiento. Este mecanismo busca evitar errores en la práctica virtual y guiar al usuario en la toma de decisiones adecuada.

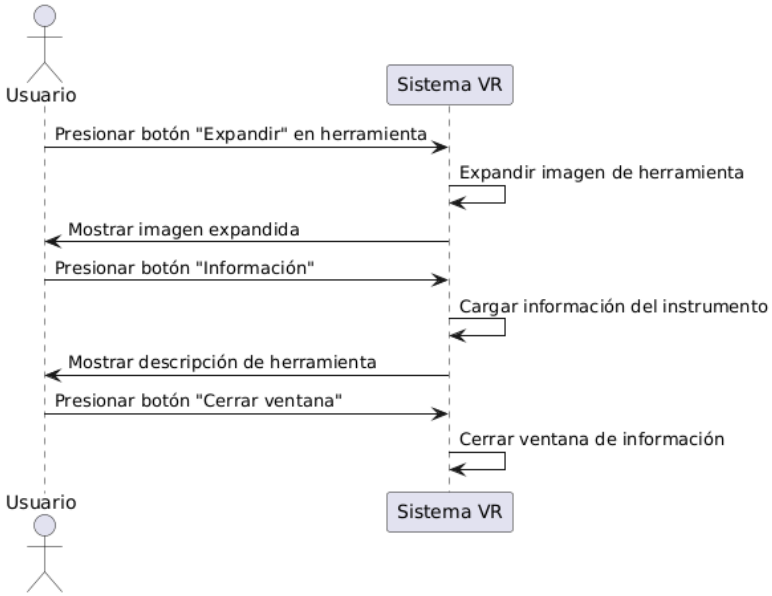


Figura 108: Diagrama de secuencia: Expansión de Imagen de Herramienta y Consulta de Información

En la Figura 108, se ilustra el diagrama de secuencia relacionado con la interacción del usuario para obtener información detallada sobre las herramientas. El proceso comienza cuando el usuario presiona el botón para expandir la imagen de una herramienta, seguido de la solicitud de información adicional. El sistema responde mostrando una descripción del instrumento seleccionado. Finalmente, el usuario puede cerrar esta ventana informativa. Este flujo mejora la experiencia educativa del entorno VR, permitiendo al usuario conocer las características de los instrumentos antes de usarlos.

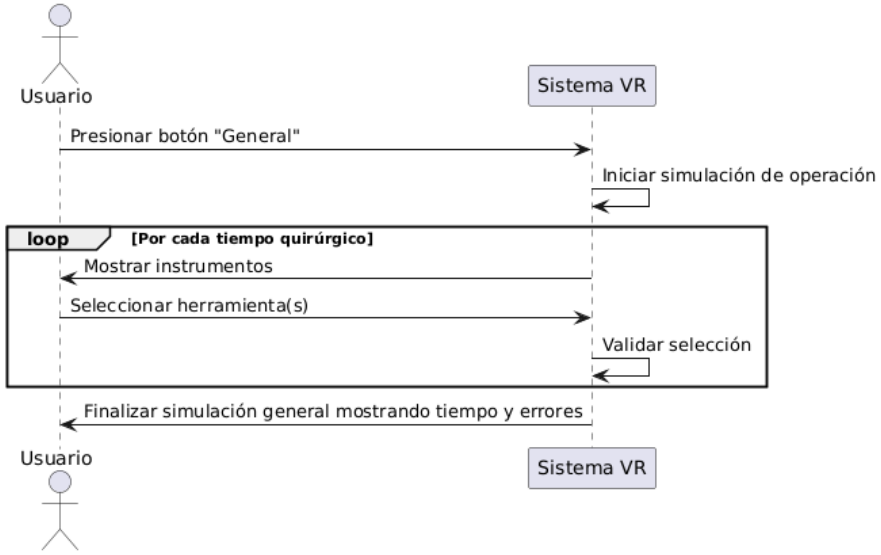


Figura 109: Diagrama de secuencia: Finalizar Simulación General

c. Codificación

Uno de los aspectos clave de la programación extrema es el contacto permanente que mantendrás con el cliente durante todo el proceso de codificación. Esta interacción te permite probar e incorporar comentarios dentro de cada iteración, en lugar de esperar hasta el final de cada sprint. Sin embargo, las reglas de codificación son bastante estrictas en el método XP. Algunas incluyen:

- Todo el código debe cumplir con el estándar de programación.
- La programación se realiza en parejas: dos desarrolladores trabajan juntos y simultáneamente en la misma computadora. El tiempo de programación sigue siendo el mismo; sin embargo, se duplica el enfoque para lograr resultados de la más alta calidad posible.
- Se deben usar integraciones continuas para agregar código nuevo y probarlo de inmediato.
- Propiedad colectiva del código: cualquier miembro del equipo puede cambiar el código en cualquier momento.

- ***Estructura de las clases del sistema***

```
using UnityEngine;

public class CanvasController : MonoBehaviour
{
    public GameObject canvasMenus;
    public GameObject btnIniciar;
    public GameObject MesaVestida;
    public GameObject MesaNormal;
    public GameObject Regresar;
    public void ActivarCanvas()
    {
        canvasMenus.SetActive(true);
        btnIniciar.SetActive(false); // Oculta el canvas actual si quieres
        MesaVestida.SetActive(true);
        MesaNormal.SetActive(false);
        Regresar.SetActive(false);
    }

    public void RegresarCanvas()
    {
        canvasMenus.SetActive(false);
        btnIniciar.SetActive(true); // Oculta el canvas actual si quieres
        MesaVestida.SetActive(false);
        MesaNormal.SetActive(true);
        Regresar.SetActive(true);
    }

}

```

Figura 110: Script CanvasController para la activación de los canvas

Como se observa en la Figura 110, la clase CanvasController, gestiona la visibilidad de distintos elementos de la interfaz de usuario en una escena. Permite activar y desactivar objetos como menús, botones y mesas, según las acciones del usuario. A través del método ActivarCanvas, se muestra un menú principal (canvasMenus), se oculta el botón de inicio (btnIniciar), se activa una mesa vestida (MesaVestida), se desactiva una mesa normal (MesaNormal) y se oculta el botón de regresar (Regresar). Por otro lado, el método RegresarCanvas revierte estos cambios, ocultando el menú, mostrando el botón de inicio, activando la mesa normal, desactivando la mesa vestida y mostrando el botón de regresar, permitiendo así alternar entre dos estados visuales de la interfaz según la interacción del usuario.

```
public class CargaAsincrona : MonoBehaviour
{
    public GameObject pantallaCarga;
    public Slider barraProgreso;
    public TMP_Text textoProgreso;
    public string nombreEscena;

    public GameObject pantallaInicio;
    public void CargarEscena()
    {
        pantallaCarga.SetActive(true);
        StartCoroutine(CargarEscenaAsync());
        pantallaInicio.SetActive(false);
    }

    IEnumerator CargarEscenaAsync()
    {
        textoProgreso.text = "Cargando...";

        // Empieza la carga de escena de forma asincrónica
        AsyncOperation operacion = SceneManager.LoadSceneAsync(nombreEscena);

        // Permite que se active automáticamente al terminar
        operacion.allowSceneActivation = false;

        while (operacion.progress < 0.9f)
        {
            float progreso = Mathf.Clamp01(operacion.progress / 0.9f);
            barraProgreso.value = progreso;
            yield return null;
        }

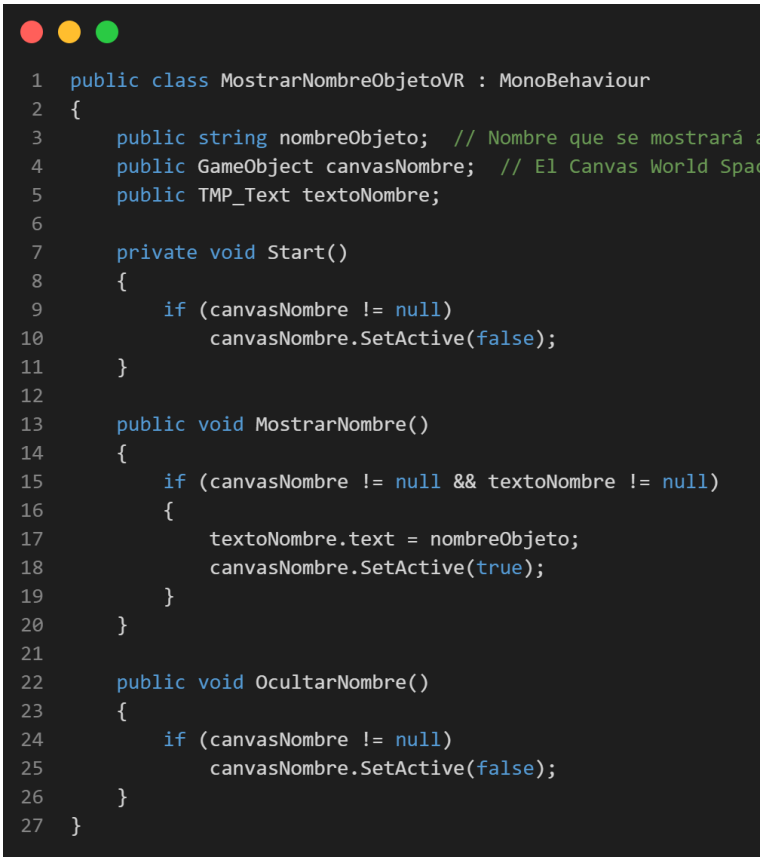
        // Escena está casi lista (en 0.9), mostrar barra llena
        barraProgreso.value = 1f;
        textoProgreso.text = "Cargando...";

        // Esperar unos 2 segundos opcionalmente para estética
        yield return new WaitForSeconds(2f);

        // Ahora sí activar la escena
        operacion.allowSceneActivation = true;
    }
}
```

Figura 111: Script CargaAsincrona para el cambio de escenarios

En la Figura 111, el script CargaAsincrona se encarga de realizar la transición entre escenas de forma fluida en Unity mediante una carga asincrónica. Al ejecutar CargarEscena(), activa una pantalla de carga con una barra de progreso y un texto informativo mientras se carga la nueva escena en segundo plano. La carga se controla manualmente para permitir mostrar una animación o mensaje por un breve periodo antes de activar la nueva escena, mejorando así la experiencia visual del usuario.



```
public class MostrarNombreObjetoVR : MonoBehaviour
{
    public string nombreObjeto;  // Nombre que se mostrará a
    public GameObject canvasNombre;  // El Canvas World Spac
    public TMP_Text textoNombre;

    private void Start()
    {
        if (canvasNombre != null)
            canvasNombre.SetActive(false);
    }

    public void MostrarNombre()
    {
        if (canvasNombre != null && textoNombre != null)
        {
            textoNombre.text = nombreObjeto;
            canvasNombre.SetActive(true);
        }
    }

    public void OcultarNombre()
    {
        if (canvasNombre != null)
            canvasNombre.SetActive(false);
    }
}
```

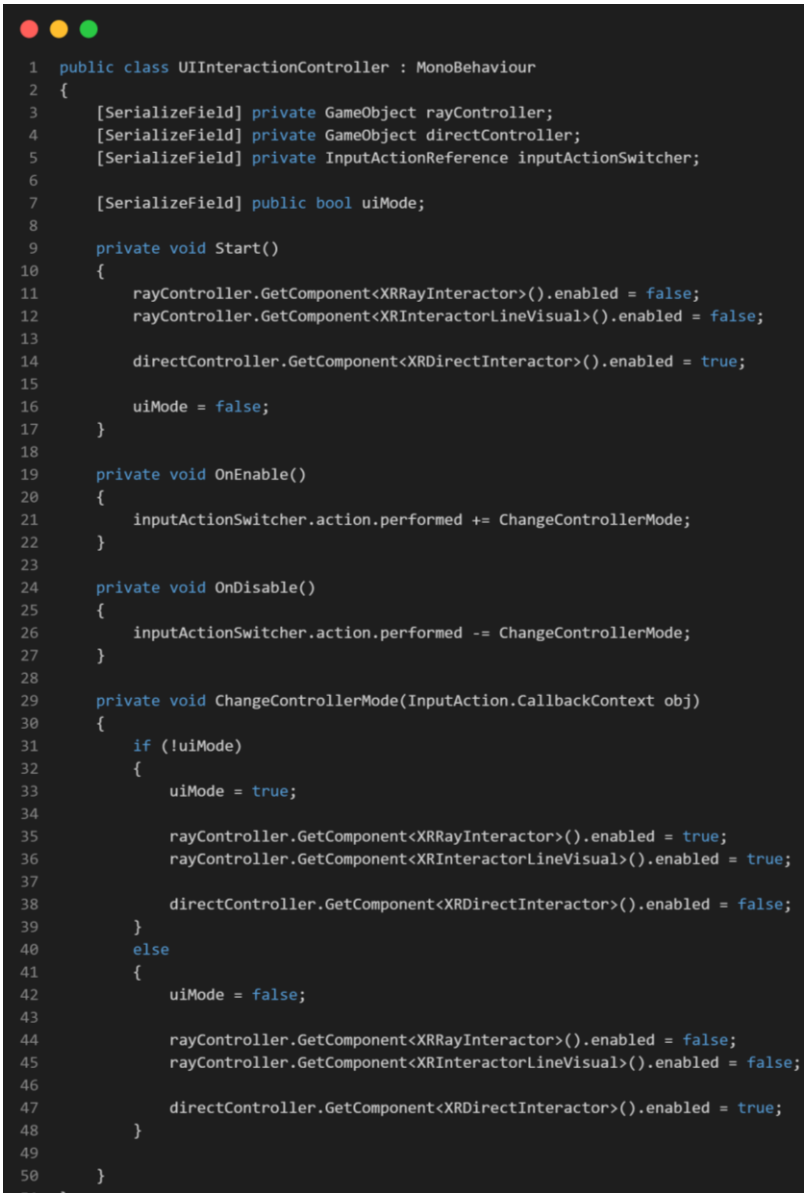
Figura 112: Script MostrarNombreObjetoVR para ver el nombre de los objetos con los punteros

Como se puede ver en la Figura 112, la clase MostrarNombreObjetoVR, permite mostrar el nombre de un objeto en un entorno de realidad virtual al momento de ser apuntado. Para ello, utiliza un canvas en espacio mundial (canvasNombre) que contiene un componente de texto (TMP_Text) donde se despliega el nombre asignado al objeto (nombreObjeto). Al iniciar la escena, el canvas se mantiene desactivado por defecto. A través del método MostrarNombre, se asigna el nombre al texto y se activa el canvas para hacerlo visible al usuario. Por el contrario, el método OcultarNombre desactiva el canvas, ocultando así la información cuando el objeto deja de ser señalado. Esta clase es útil para brindar retroalimentación visual en experiencias interactivas de realidad virtual.

```
1  public class MenuPrincipalUta : MonoBehaviour
2  {
3      public GameObject CanvaMenuUta;
4      public GameObject CanvaTutorial;
5      public GameObject TutorialMovilizacion;
6      public GameObject TutorialPuntero;
7      public GameObject TutorialInteractuar;
8      public GameObject TutorialSalir;
9
10     public void Tutorial(){
11         CanvaMenuUta.SetActive(false);
12         CanvaTutorial.SetActive(true);
13     }
14
15
16     public void tutorialPuntero(){
17         TutorialMovilizacion.SetActive(false);
18         TutorialPuntero.SetActive(true);
19
20     }
21
22     public void tutorialInteractuar(){
23         TutorialPuntero.SetActive(false);
24         TutorialInteractuar.SetActive(true);
25
26     }
27
28     public void tutorialSalir(){
29         TutorialInteractuar.SetActive(false);
30         TutorialSalir.SetActive(true);
31     }
```

Figura 113: Script MenuPrincipalUta para controlar los canvas de la primera escena

Como se evidencia en la Figura 113, la clase MenuPrincipalUta, controla la navegación entre el menú principal y una serie de tutoriales dentro de una aplicación, probablemente orientada a un entorno educativo o de simulación en realidad virtual. Esta clase permite al usuario iniciar el tutorial desde el menú principal, avanzar paso a paso por distintas secciones del mismo (movilización, uso del puntero, interacción y salida), así como retroceder entre estas etapas según se requiera. Además, ofrece funciones para regresar al menú principal, salir de la aplicación (Application.Quit()), o cargar una nueva escena denominada "Quirofano 1", posiblemente el entorno principal de simulación. En conjunto, la clase facilita una navegación intuitiva y estructurada por el sistema de aprendizaje virtual.



```
public class UIInteractionController : MonoBehaviour
{
    [SerializeField] private GameObject rayController;
    [SerializeField] private GameObject directController;
    [SerializeField] private InputActionReference inputActionSwitcher;

    [SerializeField] public bool uiMode;

    private void Start()
    {
        rayController.GetComponent<XRRayInteractor>().enabled = false;
        rayController.GetComponent<XRInteractorLineVisual>().enabled = false;

        directController.GetComponent<XRDirectInteractor>().enabled = true;

        uiMode = false;
    }

    private void OnEnable()
    {
        inputActionSwitcher.action.performed += ChangeControllerMode;
    }

    private void OnDisable()
    {
        inputActionSwitcher.action.performed -= ChangeControllerMode;
    }

    private void ChangeControllerMode(InputAction.CallbackContext obj)
    {
        if (!uiMode)
        {
            uiMode = true;

            rayController.GetComponent<XRRayInteractor>().enabled = true;
            rayController.GetComponent<XRInteractorLineVisual>().enabled = true;

            directController.GetComponent<XRDirectInteractor>().enabled = false;
        }
        else
        {
            uiMode = false;

            rayController.GetComponent<XRRayInteractor>().enabled = false;
            rayController.GetComponent<XRInteractorLineVisual>().enabled = false;

            directController.GetComponent<XRDirectInteractor>().enabled = true;
        }

    }
```

Figura 114: Script UIInteractionController para los controladores de manos

Se puede observar en la Figura 114, el script UIInteractionController permite alternar entre dos modos de interacción en una experiencia de realidad virtual usando el XR Interaction Toolkit de Unity: la interacción directa (con las manos, mediante XRDirectInteractor) y la interacción por rayo láser (para interfaces UI, usando XRRayInteractor y XRInteractorLineVisual). Utiliza un InputActionReference llamado inputActionSwitcher para detectar cuándo el usuario desea cambiar entre estos modos. Al iniciar, el script activa solo la interacción directa, desactivando el rayo. Cuando el usuario activa la acción configurada (por ejemplo, presionar un botón del control), se intercambian los modos: si estaba en interacción directa, se activa el

rayo y se desactiva la directa, y viceversa. Este sistema permite cambiar fluidamente entre manipulación directa de objetos y selección de elementos UI, mejorando la experiencia inmersiva y facilitando la navegación en entornos virtuales.

```
public class ControladorExpandir : MonoBehaviour
{
    public static ControladorExpandir Instance;

    public GameObject canvasExpandido;
    public Image imagenExpandida;

    public GameObject panelInformacion; // <- el panel de info
    public TextMeshProUGUI textoDescripcion;
       // <- el componente de texto

    private string descripcionActual = "";

    void Awake()
    {
        Instance = this;
    }

    public void MostrarImagen(Sprite imagen, string descripcion)
    {
        imagenExpandida.sprite = imagen;
        descripcionActual = descripcion;
        canvasExpandido.SetActive(true);
        panelInformacion.SetActive(false); // ocultar info hasta
    }


    public void MostrarInformacion()
    {
        textoDescripcion.text = descripcionActual;
        panelInformacion.SetActive(true);
    }

```

Figura 115: Script ControladorExpandir para la visualización del instrumental quirúrgico

En la Figura 115 se observa, la clase ControladorExpandir, escrita en C# para Unity, gestiona la visualización ampliada de imágenes junto con su información descriptiva, ideal para aplicaciones educativas o de simulación. Implementa un patrón de instancia única (Singleton) a través de la propiedad estática Instance, lo que permite acceder fácilmente a sus métodos desde otros scripts. Mediante el método MostrarImagen, se presenta una imagen expandida (usando un Sprite) en un canvas específico y se guarda su descripción correspondiente, ocultando inicialmente el panel de información textual. Luego, con el método MostrarInformacion, se muestra la descripción almacenada en un panel con texto (TextMeshProUGUI). Además, la clase incluye una

estructura InstrumentoInfo, que permite asociar una imagen con una descripción, facilitando el manejo de datos relacionados con instrumentos o elementos visuales.

Restos de las Clases

Clase: MultiplayerCreationControllerGeneral

El script MultiplayerCreationControllerGeneral en Unity controla la lógica general del menú de selección de operaciones e instrumentos en un entorno de realidad virtual. Administra la transición entre menús, el despliegue de instrumentos, el reinicio del estado de la interfaz (como botones y tiempos), y coordina componentes como el control de tiempos y la asignación de instrumentos a la mesa quirúrgica. También prepara el sistema para cada nueva simulación, asegurando que todos los elementos estén listos para la interacción del usuario de forma ordenada e intuitiva.

Clase: BotonInstrumentoGeneral

El script BotonInstrumentoGeneral controla el comportamiento de cada botón asociado a un instrumento quirúrgico en la interfaz. Al hacer clic, verifica si el instrumento es válido según la operación y el tiempo seleccionado; si es correcto, lo asigna a la mesa quirúrgica y lo desactiva visualmente para evitar repeticiones. También notifica al controlador de tiempos si la selección fue correcta o incorrecta, permitiendo registrar el desempeño del usuario dentro del sistema de simulación.

Clase: AsignarInstrumentoGeneral

El script AsignarInstrumentoGeneral se encarga de gestionar la mesa quirúrgica (mesa Mayo) en la simulación. Permite agregar visualmente los instrumentos seleccionados por el usuario, ubicándolos en el panel correspondiente según su tiempo quirúrgico. Además, ofrece una función para limpiar la mesa eliminando todos los instrumentos previamente agregados, reiniciando así el entorno para una nueva simulación o intento.

Clase: ControlTiemposGeneral

El script ControlTiemposGeneral coordina el flujo de tiempos quirúrgicos dentro de la simulación, controlando el orden en que se deben seleccionar los instrumentos según la operación elegida. Lleva un conteo del tiempo total de la actividad, registra errores

cometidos por selección incorrecta y muestra estadísticas detalladas por cada fase quirúrgica. También gestiona la lógica para avanzar entre tiempos, reiniciar el control y presentar un resumen final del rendimiento del usuario.

Clase: ControlMesaMayo

El script ControlMesaMayo se encarga de gestionar la visualización y el estado de la mesa de instrumentos quirúrgicos (Mesa Mayo) en una simulación de realidad virtual en Unity. Su función principal es activar o desactivar elementos visuales (canvas, instrumentos, botones y tipos de mesa) cuando el usuario interactúa con un Toggle.

Tiene dos métodos públicos para reaccionar al cambio de estado del Toggle:

1. **AlCambiarToggle (bool estado)**: Muestra u oculta la Mesa Mayo sin iniciar ni detener el conteo de tiempo.

2. **AlCambiarToggleGeneral (bool estado)**: Similar al anterior, pero también inicia o detiene el conteo del tiempo total de la práctica, mediante el ControlTiemposGeneral.

También incluye funciones auxiliares:

- **LimpiarMesa ()**: Elimina todos los instrumentos colocados en la mesa.

- **ActivarBotonesCheck (bool activar)**: Activa o desactiva los botones de selección de instrumentos según el estado del toggle.

- ***Historial de desarrollo (GitHub)***

Se utilizó GitHub como herramienta principal para el control de versiones del proyecto, permitiendo gestionar de manera eficiente el desarrollo colaborativo entre los integrantes del equipo. GitHub facilitó el seguimiento de los cambios realizados en el código fuente, la identificación y corrección de errores, así como la posibilidad de trabajar en diferentes ramas sin afectar la versión principal del sistema. Además, proporcionó funcionalidades clave como la documentación de cada commit, la revisión de código, la integración continua y la restauración de versiones anteriores, lo

que contribuyó significativamente a mantener la organización, trazabilidad y calidad del proyecto.

A continuación, se presentará el historial de commits, donde se detallarán únicamente aquellos que resultaron más relevantes para el desarrollo del proyecto. Estos commits reflejan las principales funcionalidades implementadas, correcciones importantes y mejoras significativas realizadas a lo largo del proceso, brindando una visión clara y concisa del avance y evolución del sistema.

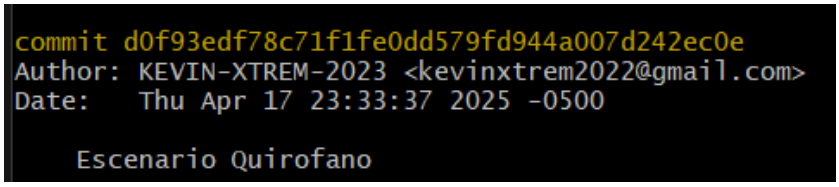


Figura 116: Commit creación del escenario del quirófano

En la Figura 116, se puede observar el primer commit realizado en el desarrollo del sistema, correspondiente a la creación inicial del escenario del quirófano. Este entorno fue diseñado para simular de manera realista el espacio donde se llevan a cabo los procedimientos médicos. La ambientación incluye elementos esenciales como cama quirúrgica, lámpara quirúrgica y equipos médicos.

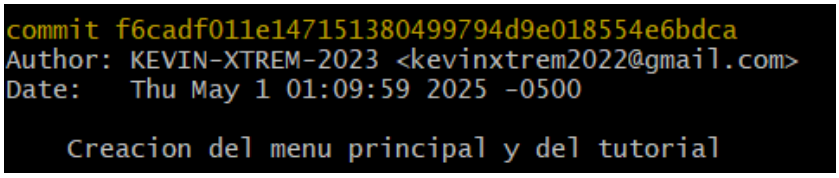


Figura 117: Commit de creación del menú principal y el Canva de tutorial

Como se aprecia en la Figura 117, este commit documenta la creación del escenario correspondiente al menú principal. Este entorno brinda acceso a información relevante del sistema, como el objetivo del proyecto, la facultad para la cual fue desarrollado y los tutoriales integrados. Estos tutoriales orientan al usuario sobre el uso correcto del sistema antes de ingresar a la simulación del quirófano. Además, se implementó una interfaz intuitiva para facilitar la navegación del estudiante.

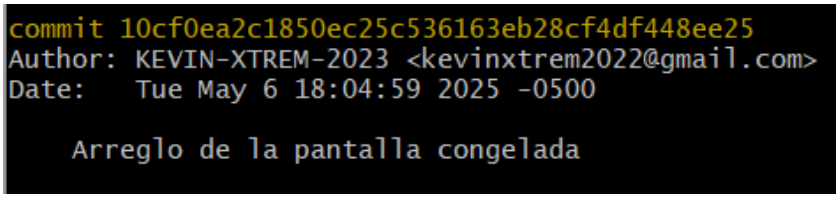


Figura 118:Commit de arreglo de la pantalla congelada al ingresar al sistema

La Figura 118, detalla el commit enfocado en la solución del problema de congelamiento de pantalla al cambiar de la escena principal a la del quirófano. Este error se originaba por la carga simultánea de múltiples recursos del entorno quirúrgico. Para solucionarlo, se optimizó el proceso de carga asincrónica, mejorando el rendimiento y evitando bloqueos. Este ajuste permitió una transición fluida entre escenas, garantizando una mejor experiencia para el usuario.

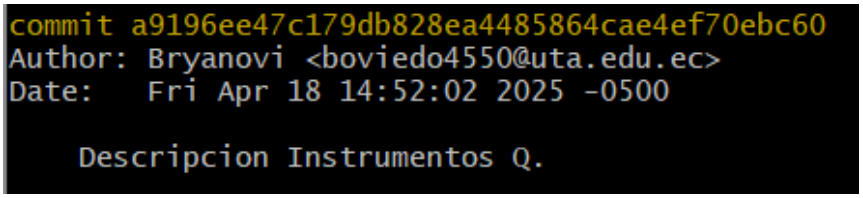


Figura 119: Commit creación del Canva Expansión de imagen

En la Figura 119, se visualiza el commit correspondiente a la creación de un Canvas que permite visualizar imágenes de los instrumentos en mayor tamaño. Además, se añadió un botón de “Ver información”, el cual despliega detalles como el nombre y características del instrumento seleccionado. Esta mejora refuerza el componente educativo del sistema, permitiendo al usuario conocer a fondo cada herramienta antes de su uso en la simulación. El diseño fue adaptado para una experiencia clara tanto en dispositivos VR como en monitores convencionales.

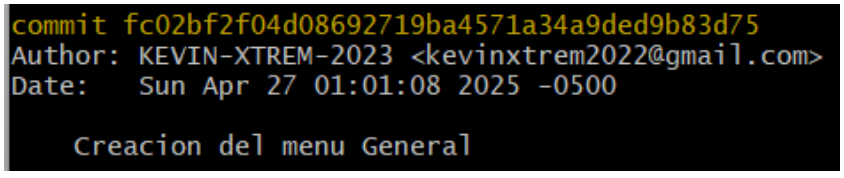


Figura 120: Commit creación de los canvas del menú de operación y del menú del instrumental quirúrgico

En la Figura 120, se observa la creación de los Canvas dedicados a los menús de operación y tiempo. Estos menús permiten seleccionar el tipo de procedimiento quirúrgico y gestionar las distintas fases de este.

d. Pruebas

Las pruebas de usabilidad son un método de evaluación que permite identificar qué tan fácil y eficiente resulta para los usuarios interactuar con un sistema, producto o aplicación. Este tipo de pruebas se centra en observar el comportamiento de los usuarios reales mientras utilizan el sistema para completar tareas específicas. A través de estas pruebas, se pueden detectar problemas de navegación, comprensión o diseño, permitiendo realizar mejoras que optimicen la experiencia del usuario. Su objetivo principal es asegurar que el sistema sea intuitivo, accesible y satisfaga las necesidades del usuario final (ver Anexo 3: Encuesta de Usabilidad).

Por otro lado, el método NASA-TLX (Task Load Index) es una herramienta desarrollada por la NASA que se utiliza para medir la carga de trabajo mental percibida por una persona al realizar una tarea. Esta técnica evalúa seis dimensiones: demanda mental, demanda física, demanda temporal, esfuerzo, rendimiento y nivel de frustración. A través de un proceso de ponderación y valoración, permite determinar qué tan exigente resulta una actividad para el usuario. Es especialmente útil en contextos donde se requiere analizar la complejidad cognitiva y física de las tareas, como en entornos de simulación o sistemas interactivos avanzados (ver Anexo 4: Cuestionario del método NASA TLX (Task Load Index)).

Toda esta información detallada sobre las pruebas de usabilidad y la aplicación del método NASA-TLX se encuentra incluida en la sección superior titulada "Métricas de Experiencia de Usuario (UX):". En dicha sección, se presentan los resultados obtenidos a partir de las encuestas aplicadas a los usuarios y el análisis de carga de trabajo, lo cual permitió evaluar de manera objetiva la facilidad de uso, comprensión del sistema y el nivel de exigencia que implicó la interacción con el entorno de realidad virtual desarrollado.

Casos de prueba de Aceptación

Los Casos de prueba de Aceptación describen escenarios específicos para probar un software con el objetivo de asegurar que cumple con los requisitos y expectativas del usuario o cliente antes de su lanzamiento.

Tabla 89: Caso de Prueba de aceptación: Navegación en el entorno virtual

Código: CPA-01	
Historia de Usuario: HU-01: Navegar por el menú principal	
Nombre:	Prueba de navegación en el entorno VR.
Descripción:	Verifica que el estudiante pueda moverse dentro del entorno virtual con los mandos de Meta Quest.
Condiciones de Ejecución:	Sistema encendido y gafas colocadas.
Entrada / Pasos de ejecución:	Iniciar aplicación Mover joystick en diferentes direcciones
Resultado Esperado:	El avatar se mueve correctamente en todas las direcciones.
Evaluación de la Prueba:	Aprobado si se observa movimiento fluido en todas las direcciones.

En la Tabla 89, se observa el caso verifica que el estudiante pueda desplazarse libremente dentro del entorno de realidad virtual utilizando los mandos de Meta Quest, asegurando un movimiento fluido en todas las direcciones.

En la Tabla 90, se valida que el estudiante pueda ingresar al menú de tutorial antes de acceder al entorno quirúrgico, permitiéndole visualizar correctamente las instrucciones y controles del sistema.

Tabla 90: Caso de Prueba de aceptación: Navegación en el entorno virtual

Código: CPA-02	
Historia de Usuario: HU-02: Navegar por el menú del tutorial	
Nombre:	Acceso al menú de tutorial.
Descripción:	Verifica que el estudiante pueda acceder al tutorial antes del entorno quirúrgico.
Condiciones de Ejecución:	Sistema activo con gafas y controladores.
Entrada / Pasos de ejecución:	Entrar al entorno virtual Seleccionar opción de tutorial Observar y seguir el tutorial
Resultado Esperado:	El estudiante visualiza correctamente las instrucciones del tutorial.
Evaluación de la Prueba:	Aprobado si se puede acceder al tutorial y se muestran los controles.

Tabla 91: Caso de Prueba de aceptación: Exploración del entorno quirúrgico

Código: CPA-03	
Historia de Usuario: HU-03: Navegar por el escenario quirúrgico	
Nombre:	Exploración del entorno quirúrgico.
Descripción:	Verifica la movilidad libre dentro del escenario quirúrgico.
Condiciones de Ejecución:	Gafas puestas y entorno cargado.
Entrada / Pasos de ejecución:	Moverse por el escenario Observar el entorno clínico
Resultado Esperado:	El escenario se muestra y es navegable.
Evaluación de la Prueba:	Aprobado si se puede recorrer el entorno quirúrgico sin errores.

Como puede verse en la Tabla 91, este caso asegura que el estudiante tenga libertad de movimiento dentro del escenario quirúrgico, permitiéndole recorrer el entorno clínico sin restricciones ni errores.

De acuerdo con la Tabla 92, se verifica que el usuario pueda elegir correctamente una operación desde un menú desplegable y que dicha selección se refleje adecuadamente en la interfaz del sistema.

Tabla 92: Caso de Prueba de aceptación: Selección del tipo de operación quirúrgica

Código: CPA-04	
Historia de Usuario: HU-04: Selección de operación	
Nombre:	Prueba de dropdown de operación.
Descripción:	Verifica que el usuario pueda seleccionar el tipo de operación desde el menú.
Condiciones de Ejecución:	Menú desplegable disponible.
Entrada / Pasos de ejecución:	Seleccionar una operación quirúrgica Presionar botón continuar
Resultado Esperado:	La operación se guarda y se muestra en el menú siguiente.
Evaluación de la Prueba:	Aprobado si la operación seleccionada se refleja correctamente.

La Tabla 93, refleja el caso asegura que el usuario pueda seleccionar un tiempo quirúrgico desde un menú desplegable, mostrando en pantalla la opción seleccionada de forma precisa.

Tabla 93: Caso de Prueba de aceptación: Elección del tiempo quirúrgico

Código: CPA-05	
Historia de Usuario: HU-05: Selección de tiempos quirúrgicos	
Nombre:	Selección correcta del tiempo quirúrgico.
Descripción:	Asegura que el usuario elija un tiempo quirúrgico válido desde el dropdown.
Condiciones de Ejecución:	Dropdown funcional.
Entrada / Pasos de ejecución:	Seleccionar tiempo quirúrgico desde el menú desplegable
Resultado Esperado:	El tiempo seleccionado aparece correctamente en la interfaz.
Evaluación de la Prueba:	Aprobado si el tiempo se refleja en pantalla.

Tabla 94: Caso de Prueba de aceptación: Asignación de instrumentos quirúrgicos

Código: CPA-06	
Historia de Usuario: HU-06: Selección de instrumentos quirúrgicos	
Nombre:	Asignación de instrumentos por tiempo.
Descripción:	Valida la selección de instrumentos válidos según el tipo de operación y tiempo.
Condiciones de Ejecución:	Toggle activado y botones habilitados.
Entrada / Pasos de ejecución:	Seleccionar instrumento correcto Verificar posicionamiento en mesa
Resultado Esperado:	El instrumento se posiciona correctamente en el panel.
Evaluación de la Prueba:	Aprobado si solo se aceptan instrumentos válidos.

Como se detalla en la Tabla 94, verifica que el usuario pueda seleccionar instrumentos válidos en función del tipo de operación y del tiempo quirúrgico, y que estos se posicionen correctamente en el panel correspondiente.

Tabla 95: Caso de Prueba de aceptación: Activación de la mesa Mayo

Código: CPA-07	
Historia de Usuario: HU-07: Activación de la mesa mayo	
Nombre:	Verificar toggle y canvas relacionados.
Descripción:	Al activar el toggle deben mostrarse el canvas de mesa y errores.
Condiciones de Ejecución:	Toggle presente y canvas ocultos.
Entrada / Pasos de ejecución:	Activar toggle de mesa
Resultado Esperado:	Ambos canvas aparecen correctamente.
Evaluación de la Prueba:	Aprobado si toggle activa los paneles requeridos.

La Tabla 95, se observa el caso comprueba que, al activar el interruptor correspondiente, se muestren correctamente los paneles visuales asociados a la mesa Mayo y los mensajes de error.

Tabla 96: Caso de Prueba de aceptación: Sistema de advertencias por errores

Código: CPA-08	
Historia de Usuario: HU-08: Verificación de errores	
Nombre:	Validación del sistema de advertencias.
Descripción:	Asegura que los errores se acumulen correctamente al elegir instrumentos inválidos.
Condiciones de Ejecución:	Canvas errores activo.
Entrada / Pasos de ejecución:	Seleccionar instrumento incorrecto
Resultado Esperado:	Se incrementa el contador de errores.
Evaluación de la Prueba:	Aprobado si el mensaje de advertencia aparece.

De acuerdo con la Tabla 96, se valida que, al seleccionar instrumentos incorrectos, el sistema genere advertencias visibles y actualice el contador de errores de manera adecuada.

La Tabla 97, verifica que, al completar todos los tiempos quirúrgicos, se muestre el tiempo total de ejecución en el panel correspondiente.

Tabla 97: Caso de Prueba de aceptación: Visualización del tiempo total de armado

Código: CPA-09	
Historia de Usuario: HU-09: Mostrar tiempo total	
Nombre:	Cronometría completa del armado.
Descripción:	Valida que se muestre el tiempo total de ejecución al finalizar todos los tiempos.
Condiciones de Ejecución:	Todos los tiempos completados.
Entrada / Pasos de ejecución:	Finalizar armado completo
Resultado Esperado:	El tiempo total aparece en el panel de errores.
Evaluación de la Prueba:	Aprobado si el cronómetro total se actualiza correctamente.

Como se observa en Tabla 98, verifica que, al concluir todos los tiempos quirúrgicos, el sistema despliegue un mensaje indicando la finalización completa de la actividad simulada.

Tabla 98: Caso de Prueba de aceptación: Mensaje de finalización de la actividad

Código: CPA-10	
Historia de Usuario: HU-10: Mensaje de finalización	
Nombre:	Confirmación de cierre de actividad.
Descripción:	Verifica que se muestre el mensaje de finalización al completar todos los tiempos.
Condiciones de Ejecución:	Simulación completa.
Entrada / Pasos de ejecución:	Finalizar último tiempo quirúrgico
Resultado Esperado:	Mensaje de finalización se despliega en pantalla.
Evaluación de la Prueba:	Aprobado si se muestra 'Usted ha terminado todos los tiempos'.

Tabla 99: Caso de Prueba de aceptación: Reinicio completo del escenario quirúrgico

Código: CPA-11	
Historia de Usuario: HU-11: Reinicio de escena	
Nombre:	Reinicio completo del escenario quirúrgico.
Descripción:	Verifica que al volver al menú y comenzar otra operación, los datos de ejecución anteriores se limpian.
Condiciones de Ejecución:	Escena completada y menú principal accesible.
Entrada / Pasos de ejecución:	Finalizar simulación actual Volver al menú principal Dar clic en botón General
Resultado Esperado:	Los errores, tiempos e instrumentos son reiniciados.
Evaluación de la Prueba:	Aprobado si todo se reinicia al comenzar una nueva operación.

De acuerdo con la Tabla 99 este caso comprueba que, al volver al menú principal e iniciar una nueva operación, todos los datos anteriores (instrumentos, tiempos y errores) sean eliminados correctamente.

Tabla 100: Caso de Prueba de aceptación: Aleatorización del orden de los instrumentos

Código: CPA-12	
Historia de Usuario: HU-12: Desordenar imágenes de instrumentos	
Nombre:	Aleatorización del contenido del scroll view.
Descripción:	Valida que cada vez que se entra al menú de instrumentos, las imágenes cambian de posición.
Condiciones de Ejecución:	Menú instrumentos con scroll view activo.
Entrada / Pasos de ejecución:	Ingresar desde menú principal a General Observar el orden de las imágenes
Resultado Esperado:	Las imágenes aparecen en distinto orden cada vez.
Evaluación de la Prueba:	Aprobado si el orden no se repite siempre igual.

De acuerdo con la Tabla 100, se valida que cada vez que el usuario acceda al menú de instrumentos, el orden de las imágenes sea diferente, garantizando una presentación no predecible.

Tabla 101: Caso de Prueba de aceptación: Identificación de objetos con puntero láser

Código: CPA-13	
Historia de Usuario: HU-13: Visualizar mensaje de nombre de objeto apuntado	
Nombre:	Identificación de objetos por puntero láser.
Descripción:	Verifica que al apuntar con el puntero se muestre el nombre del objeto, y al quitarlo desaparezca.
Condiciones de Ejecución:	Gafas VR activas con raycaster.
Entrada / Pasos de ejecución:	Apuntar a un objeto interactivo como silla Observar aparición del nombre Retirar puntero
Resultado Esperado:	El nombre aparece y desaparece según se apunte.
Evaluación de la Prueba:	Aprobado si el texto aparece solo al apuntar y se oculta después.

La Tabla 101 verifica que, al apuntar a un objeto con el puntero láser, se muestre su nombre en pantalla, y que dicho nombre desaparezca al dejar de apuntarlo.

e. Lanzamiento

En la fase de lanzamiento del sistema desarrollado en XP (Programación Extrema), se concluyó exitosamente, cumpliendo con todos los requisitos establecidos y validando su correcto funcionamiento. El producto final, está listo para ser utilizado por los usuarios finales, habiendo superado las pruebas correspondientes y asegurando una experiencia inmersiva y estable. Para quienes deseen instalar y probar el sistema, pueden acceder al siguiente enlace que los dirigirá al archivo APK de instalación: https://drive.google.com/drive/folders/1YsEsRyWWQXMQEUMMZQ85Y2KXl6UmtPKv?usp=sharing.

Diseño de la metodología para desarrollo de aplicaciones con realidad virtual para enfermería

- En esta sección, los investigadores justificamos científicamente las fases propuestas en la metodología: Investigación y análisis, diseño, desarrollo, pruebas, implantación y mantenimiento, en el contexto de desarrollo de

aplicaciones de RV en salud, mediante la identificación de prácticas, enfoques y modelos validados en la literatura científica indexada en la base de datos SCOPUS.

Cadena de Búsqueda (Scopus):

- La cadena de búsqueda utilizada por los investigadores fue ("virtual reality" OR "VR") AND ("software development" OR "software engineering") AND ("nursing" OR "surgery") AND ("methodology" OR "development process" OR "life cycle") AND ("design" OR "implementation" OR "testing" OR "maintenance")

Se aplicaron los siguiente filtros:

- Periodo: 2015–2025
- Tipo de documento: Article, Conference Paper, Review
- Idioma: English
- Áreas: Computer Science, Engineering, Health Professions, Medicine

Criterios de Inclusión y Exclusión

Los criterios de Inclusión utilizados fueron:

- Estudios con aplicaciones de RV en contextos clínicos
- Publicaciones que aborden fases del desarrollo de software
- Investigaciones que involucren diseño instruccional o UX
- Trabajos con validación en entornos reales (nursing/surgery)

Los criterios de exclusión fueron:

- Proyectos de RV en contextos puramente lúdicos o artísticos
- Estudios sin enfoque metodológico o sin fases definidas

- Artículos de opinión sin respaldo empírico
- Publicaciones sin validación empírica

Para esta sección de la investigación se utilizó la metodología PRISMA en vista que proporciona una guía estructurada para realizar y reportar revisiones sistemáticas.

A continuación, detallamos brevemente las etapas del flujo de revisión sistemática:

- Registros identificados (Scopus): 315
- Duplicados eliminados: 45
- Evaluados por título/resumen: 270
- Evaluados por texto completo: 63

Incluidos en la revisión final: 28 artículos

Análisis de resultados

La revisión sistemática realizada permitió identificar una serie de estudios clave que respaldan y justifican la inclusión de cada una de las fases en la metodología de desarrollo de aplicaciones de realidad virtual (RV) aplicada a contextos clínicos, particularmente en enfermería y cirugía. Estos estudios no solo validan la estructura cíclica propuesta, sino que también aportan evidencia empírica sobre la necesidad de abordar cada fase de forma estructurada y con participación activa de los usuarios finales, tales como médicos, desarrolladores y estudiantes. En primer lugar, la fase de 'Investigación y Análisis' se ve ampliamente respaldada por el estudio de [102], el cual se enfoca en la ingeniería de requerimientos centrada en el usuario para entornos de entrenamiento quirúrgico. Este trabajo subraya la importancia de realizar una recopilación sistemática de necesidades desde el inicio, incorporando la experiencia de los profesionales de la salud para garantizar que los escenarios desarrollados en RV reflejen con precisión la complejidad y las condiciones reales del entorno clínico. Sin una fase sólida de análisis, los proyectos corren el riesgo de producir soluciones técnicamente viables, pero clínicamente irrelevantes. La fase de 'Diseño' es abordada

por [103], quienes presentan un enfoque de diseño de experiencia de usuario (UX) para entornos virtuales inmersivos destinados a la educación en enfermería. Este estudio destaca cómo un diseño instruccional centrado en el usuario puede mejorar significativamente la inmersión, la comprensión y la retención de habilidades clínicas. La participación activa de los estudiantes en el diseño, así como la inclusión de elementos pedagógicos como retroalimentación inmediata y escenarios adaptativos, son claves en esta fase. El diseño no solo se refiere al aspecto visual o técnico, sino a cómo se estructura la interacción entre el usuario y la simulación, lo cual tiene un impacto directo en los resultados de aprendizaje. En cuanto a la fase de 'Desarrollo', en [104] analizan la arquitectura de software y los desafíos técnicos en la creación de un sistema de entrenamiento en cirugía mínimamente invasiva basado en RV. Los autores detallan cómo se lleva a cabo la implementación efectiva de los módulos RV, incluyendo la sincronización de datos, la simulación de tejidos blandos y la respuesta háptica. Este estudio pone en evidencia la necesidad de contar con una fase de desarrollo que sea iterativa, flexible y colaborativa, permitiendo integrar mejoras continuas a partir de las pruebas y la retroalimentación clínica. La fase de 'Pruebas' encuentra respaldo empírico en el trabajo de [105], quienes evaluaron la fidelidad y usabilidad de una simulación de RV para emergencias en enfermería. Este estudio se enfoca en medir tanto la percepción del usuario como el rendimiento técnico del sistema, utilizando métricas como él SUS (System Usability Scale) y evaluaciones observacionales. Los resultados indican que la validación técnica y educativa del simulador es indispensable para garantizar que las herramientas desarrolladas no solo funcionen correctamente, sino que también generen impacto positivo en el aprendizaje clínico. Se destaca además la importancia de realizar pruebas con distintos grupos de usuarios (expertos, novatos y estudiantes) para identificar problemas de accesibilidad, navegabilidad o realismo clínico. La fase de 'Implantación' es discutida en profundidad por [106], quienes documentan la transición desde el prototipo hasta la aplicación práctica de herramientas de RV en entrenamiento hospitalario. Este estudio subraya los retos logísticos, administrativos y tecnológicos que deben enfrentarse durante la introducción de nuevas tecnologías en entornos clínicos reales. La implantación exitosa no depende únicamente del desarrollo técnico, sino también de la aceptación institucional, la formación del personal y la compatibilidad con los planes de estudio o protocolos hospitalarios existentes. La fase de implantación incluye pruebas piloto,

capacitaciones, y monitoreo inicial del uso del sistema para asegurar su efectividad y sostenibilidad. Finalmente, la fase de 'Mantenimiento' es justificada por el trabajo de [107], quienes abordan la necesidad de mantener plataformas clínicas de entrenamiento basadas en RV a lo largo del tiempo. Los autores exploran aspectos como la actualización de contenidos clínicos, la corrección de errores técnicos, la escalabilidad del sistema y la adaptación a nuevos contextos o normativas. El mantenimiento no debe ser visto como una fase final, sino como una práctica continua que garantiza la vigencia y evolución del sistema. Este componente es especialmente importante en entornos médicos, donde los protocolos y tecnologías cambian constantemente, requiriendo que las plataformas RV se mantengan alineadas con la realidad clínica actual.

3.9 Metodología planteada para el Desarrollo de Realidad Virtual

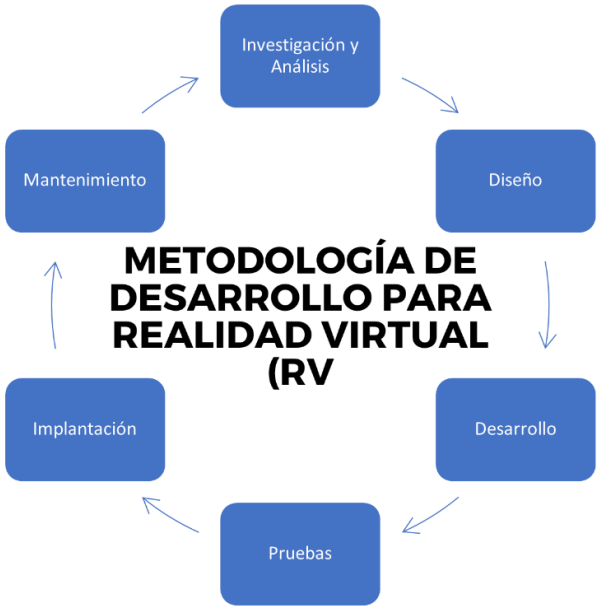


Figura 121: Metodología para el Desarrollo de Aplicaciones de Realidad Virtual (RV) para Escenarios de Enfermería en Cirugía

La Figura 121, muestra una metodología cíclica que se planteó para el desarrollo de aplicaciones de Realidad Virtual (RV), compuesta por seis etapas: investigación y análisis, diseño, desarrollo, pruebas, implantación y mantenimiento. Esta metodología está enfocada en escenarios de enfermería en cirugía.

3.9.1 Fase de Investigación y Análisis

La primera fase del desarrollo de aplicaciones de realidad virtual (RV) para escenarios de enfermería en cirugía es la investigación y análisis, donde se identifican los requisitos funcionales y no funcionales del sistema ver Especificación de Requerimientos de Software (SRS) basado en la norma ISO 29148:2018. Los requisitos funcionales incluyen la simulación de procedimientos quirúrgicos específicos, como la preparación del quirófano, el manejo de instrumentos y la asistencia durante la cirugía, mientras que los requisitos no funcionales abarcan la precisión médica, la baja latencia y la compatibilidad con hardware de RV existente. Este proceso puede ser desafiante debido a la necesidad de equilibrar la precisión médica con las limitaciones técnicas. Además, se realiza un estudio de viabilidad técnica para evaluar si las tecnologías actuales pueden cumplir con estos requisitos, analizando motores de RV como Unity o Unreal Engine, herramientas de modelado 3D y dispositivos hápticos para simular tacto y manipulación de instrumentos. La integración de múltiples tecnologías puede aumentar la complejidad y los costos. Finalmente, se fomenta la colaboración interdisciplinaria entre ingenieros de software, diseñadores de RV, expertos en cirugía y profesionales de enfermería quirúrgica para garantizar que el software sea técnicamente robusto y médicamente preciso, aunque la comunicación entre disciplinas puede ser un reto.

Roles:

- **Investigador Principal:** Lidera la identificación de requisitos funcionales y no funcionales (Daniel Sánchez – Kevin Saquinga).
- **Expertos Médicos:** Cirujanos, enfermeras quirúrgicas y especialistas en anatomía que validan la precisión médica (Dr. José Luis Herrera López – Dra. Marianela Mejías de Duarte – Dra. Yeisy Cristina Guarate Coronado – Mg. Ana Pamela Pachucho Flores).
- **Analista de Sistemas:** Evalúa la viabilidad técnica de las tecnologías disponibles (Daniel Sánchez – Kevin Saquinga).

- **Diseñador Instruccional:** Define los objetivos pedagógicos y alinea la metodología con los resultados buscados (Daniel Sánchez – Kevin Saquinga).

Entregable

- Documento de requisitos funcionales y no funcionales (ver Análisis de requerimientos).

- Informe de viabilidad técnica (evaluación de motores de RV, lenguajes de programación, gafas de realidad virtual y herramientas de modelado) para más detalles (ver Tabla 1. Motores de desarrollo VR, Tabla 2. Lenguajes de programación comunes en VR, Tabla 4. Herramientas de modelado 3D).

3.9.2 Fase de Diseño

En la fase de diseño, se establece una arquitectura modular que facilite la escalabilidad, el mantenimiento y la integración de futuras funcionalidades en la aplicación de realidad virtual. El sistema se estructura en componentes clave como el motor de simulación, el módulo de interacción, la base de datos de escenarios quirúrgicos y la interfaz de usuario. Se adopta un enfoque basado en el patrón Modelo-Vista-Controlador (MVC) para separar la lógica de simulación del entorno visual y de los controles del usuario.

Durante esta etapa, se definen los escenarios clínicos a simular, centrados en procedimientos quirúrgicos básicos como la apendicetomía y la cesárea. Para cada procedimiento, se diseñan secuencias de pasos representadas mediante storyboards y flujos de interacción, con énfasis en el uso correcto de instrumentos y la comprensión del entorno quirúrgico. Estos diseños consideran elementos visuales e interactivos que orientan al estudiante durante el proceso, como etiquetas flotantes con descripciones de herramientas, retroalimentación visual al seleccionar instrumentos y sistemas de validación del orden de uso.

Además, se planifica la integración de librerías específicas para locomoción del usuario en el entorno virtual, utilizando Final IK para permitir desplazamiento mediante controladores. Se definen zonas de interacción y navegación dentro de la sala

quirúrgica virtual, con la finalidad de fomentar la exploración del entorno y la correcta ubicación de instrumentos.

Se elabora también el diseño inicial de la interfaz de usuario (UI), priorizando la claridad, accesibilidad y orientación educativa. Se incluyen menús contextuales, paneles de información sobre instrumentos, y una navegación sencilla por los distintos escenarios disponibles.

Esta fase implica un trabajo detallado de planificación tanto gráfica como funcional, anticipando los desafíos técnicos del desarrollo y buscando un balance entre realismo, interactividad y viabilidad técnica.

Roles

- **Arquitecto de Software:** Diseña la arquitectura modular del sistema (Daniel Sánchez – Kevin Saquinga).
- **Diseñador de Experiencia de Usuario (UX):** Crea storyboards y prototipos de baja fidelidad (Daniel Sánchez – Kevin Saquinga).
- **Modelador 3D:** Se desarrollaron los inmuebles que conforman un quirófano. En cuanto al instrumental quirúrgico, se utilizó únicamente fotografías, ya que modelar elementos como las tijeras resulta complejo debido al alto nivel de precisión que requieren (Daniel Sánchez – Kevin Saquinga).

Entregable

- Diagramas de caso de uso (ver Figura 81: Diagrama de casos de uso, Figura 82: Diagrama de casos de uso: Flujo completo de navegación entre menús, Figura 83: Diagrama de casos de uso: Simulación de Operación General, Figura 84: Diagrama de casos de uso: Seleccionar Herramienta Correcta/Incorrecta).
- Historias de usuario (ver Tabla 50: Historia de Usuario: Navegar por el menú principal, Tabla 51: Historia de Usuario: Navegar por el menú del tutorial, Tabla 52: Historia de Usuario: Navegar por el escenario de simulación quirúrgico, Tabla 53: Historia de Usuario: Selección de operación, Tabla 54:

Historia de Usuario: Selección de tiempos quirúrgicos, Tabla 55: Historia de Usuario: Selección de los instrumentos quirúrgicos, Tabla 56: Historia de Usuario: Selección de los instrumentos quirúrgicos, Tabla 57: Historia de Usuario: Selección de los instrumentos quirúrgicos, Tabla 58: Historia de Usuario: Mostrar tiempo total, Tabla 59: Historia de Usuario: Mensaje de finalización, Tabla 60: Historia de Usuario: Reiniciar el sistema al volver, Tabla 61: Historia de Usuario: Interacción XR con objetos, Tabla 62: Historia de Usuario: Aleatorización de instrumentos).

- Prototipo de baja fidelidad de la interfaz de usuario (ver Diseño de prototipado en Figma

3.9.3 Fase de Desarrollo

La fase de desarrollo comienza con el prototipado rápido, creando versiones de baja fidelidad para validar conceptos antes de invertir recursos en el desarrollo completo. Estos prototipos se prueban con profesionales de enfermería quirúrgica para obtener retroalimentación temprana, aunque pueden no capturar completamente la complejidad de los escenarios quirúrgicos. Posteriormente, se sigue un enfoque ágil (Scrum, Kanban, XP) para el desarrollo iterativo, dividiendo el proyecto en sprints enfocados en funcionalidades específicas, como la simulación del armado del instrumental quirúrgico en la mesa mayo para las operaciones de cesárea y apendicetomía, y realizando revisiones regulares con stakeholders para asegurar alineación con los objetivos. Mantener el equilibrio entre velocidad de desarrollo y calidad puede ser un desafío. Finalmente, se integra contenido médico preciso y relevante, colaborando con cirujanos y expertos en procedimientos quirúrgicos para validar modelos 3D (Inmuebles de un quirófano) y simulaciones, e incorporando casos clínicos basados en estudios reales para aumentar la autenticidad, aunque este proceso puede ser lento y costoso.

Roles:

- **Desarrollador de Software:** Implementa el motor de simulación y las funcionalidades principales (Daniel Sánchez – Kevin Saquinga).

- **Programador de RV:** Optimiza gráficos y recursos para dispositivos de RV (Daniel Sánchez – Kevin Saquinga).

- **Experto en Contenido Médico:** Supervisa la integración de casos clínicos y validación de contenido (Dr. José Luis Herrera López – Dra. Marianela Mejías de Duarte – Dra. Yeisy Cristina Guarate Coronado – Mg. Ana Pamela Pachucho Flores).

Entregable

- Versión beta funcional

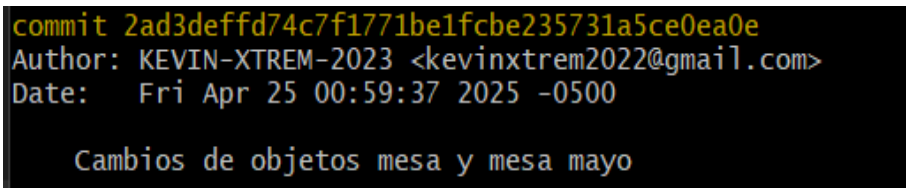


Figura 122: Commit de la versión beta Funcional

Como se observa en la Figura 122, se presenta la versión beta funcional del sistema, la cual incluye únicamente el escenario del quirófano. En este entorno, el estudiante tiene la posibilidad de seleccionar el tipo de operación junto con su respectivo tiempo, y podrá armar la mesa Mayo. Sin embargo, en esta versión solo se encuentra habilitada la parte de entrenamiento, la cual consiste en que el estudiante puede armar el tiempo quirúrgico que haya seleccionado.



Figura 123: Escenario del quirófano diseñado para su versión beta

Se puede observar en la Figura 123, el escenario del quirófano que fue presentado en la primera versión beta del sistema. Sin embargo, dado que se está utilizando la metodología XP, fue necesario presentar prototipos del sistema, lo que permitió recibir retroalimentación con respecto a algunos cambios a implementar.



Figura 124: Menú de entrenamiento para la identificación del instrumental quirúrgico

En la Figura 124, se puede observar el menú de entrenamiento correspondiente a la versión beta del sistema, una vez presentado este módulo, se recibió retroalimentación que indicó la necesidad de realizar ciertos cambios. Entre las sugerencias se incluyeron

la mejora de los fondos de los instrumentales y la implementación de un mecanismo para validar el entrenamiento.

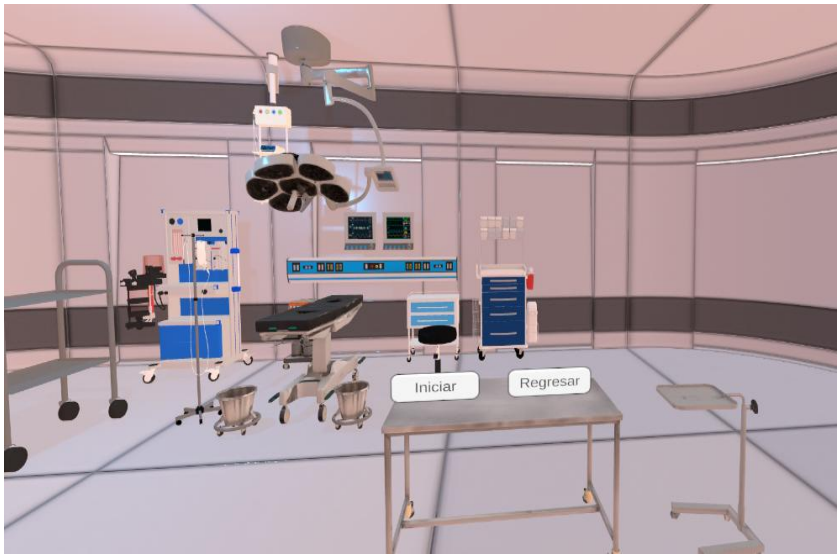


Figura 125: Módulo principal la interacción del instrumental quirúrgico

Como se observa en la Figura 125, se presenta el módulo principal del simulador. Una vez que el usuario interactúa con los botones disponibles, se procede a realizar el procedimiento quirúrgico del armado de la mesa Mayo, dependiendo de los parámetros que haya seleccionado previamente.



Figura 126: Módulo de entrenamiento del sistema

Según se aprecia en la Figura 126, se muestra el módulo de entrenamiento. En este módulo, el usuario únicamente puede armar el tiempo quirúrgico que haya escogido

con anterioridad, en función del tipo de operación y del tiempo seleccionado. Este módulo permite que el usuario entrene en un tiempo específico y, posteriormente, configure el sistema para practicar otro tiempo distinto.



Figura 127: Módulo de retroalimentación del instrumental

Como se aprecia en la Figura 127 se presenta el módulo de retroalimentación, el cual permite al estudiante conocer en mayor detalle el nombre de cada instrumental, junto con sus características principales.



Figura 128: Módulo de evaluación dependiendo del tipo de operación que seleccione

La Figura 128 muestra el módulo de evaluación. Aunque su apariencia es similar a la del módulo de entrenamiento mostrado en la Figura 126, su funcionamiento es

diferente. Mientras que en el módulo de entrenamiento el usuario arma la mesa Mayo únicamente para el tiempo seleccionado, en el módulo de evaluación, una vez finalizado el primer tiempo, se avanza automáticamente al siguiente, y así sucesivamente hasta completar todos los tiempos quirúrgicos.

En la Figura 129 se observa el módulo de resultados, el cual se activa al finalizar el armado completo de la mesa Mayo. Este módulo proporciona un resumen detallado que incluye la cantidad de errores cometidos por cada tiempo y el tiempo total que se tomó en completar el procedimiento.

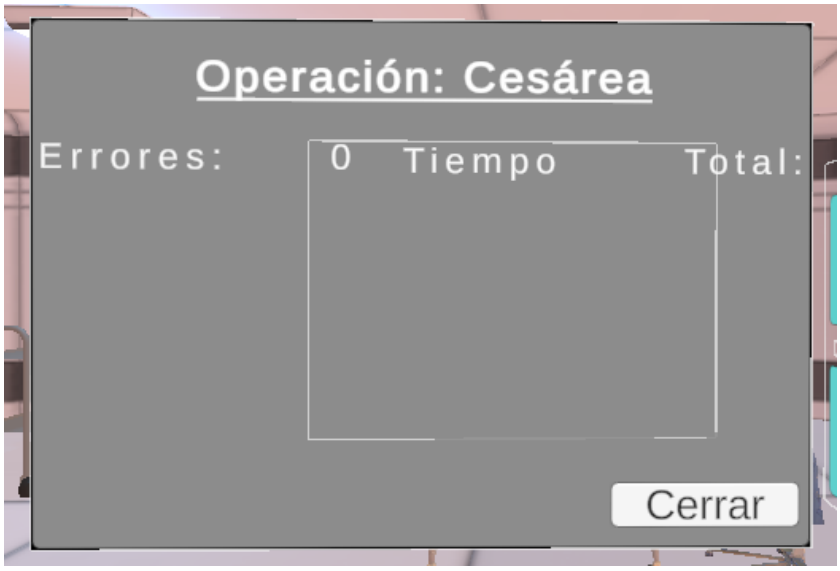


Figura 129: Módulo de resultados una vez terminado la evaluación

3.9.4 Fase de Pruebas

En la fase de pruebas, se realizan pruebas de usabilidad para garantizar que la aplicación sea fácil de usar, identificando problemas de navegación o interacción y evaluando la ergonomía de los dispositivos de RV. Las pruebas de usabilidad en RV pueden ser más complejas que en aplicaciones tradicionales debido a la inmersión. A continuación, se realizan pruebas de funcionalidad para verificar que todas las funciones operen correctamente, probando escenarios completos para detectar errores o inconsistencias y validando la precisión de las simulaciones quirúrgicas, que son inherentemente complejas y pueden presentar fallos difíciles de diagnosticar.

Finalmente, se realizan pruebas de rendimiento para asegurar que la aplicación funcione sin problemas en diferentes dispositivos, evaluando tiempos de carga, latencia y estabilidad, y optimizando gráficos y recursos para mejorar la experiencia del usuario, aunque el rendimiento es crítico en RV y puede requerir sacrificios en la calidad visual.

Roles:

- **Tester de Software:** Realiza pruebas de funcionalidad y rendimiento (Daniel Sánchez – Kevin Saquinga).
- **Profesionales Médicos:** Evalúan la precisión médica de los escenarios (Dr. José Luis Herrera López – Dra. Marianela Mejías de Duarte – Dra. Yeisy Cristina Guarate Coronado – Mg. Ana Pamela Pachucho Flores).
- **Usuario Final (Estudiantes/Docentes):** Prueban la usabilidad y proporcionan retroalimentación (Estudiantes de quinto y sexto semestre).
- **Analista de Datos:** Recopila y analiza métricas de uso y desempeño (Daniel Sánchez – Kevin Saquinga).

Entregables:

- Informe de pruebas de usabilidad: SUS ver Encuesta SUS para medir la usabilidad y NASA TLX ver Resultados de NASA-TLX
- Reporte de errores detectados y soluciones propuestas: Los errores que se identificaron durante el desarrollo del sistema estuvieron relacionados con la movilidad del avatar dentro del escenario, la posición de la cama y el diseño del mobiliario de un quirófano. Para resolver los problemas de movilidad y posicionamiento de la cámara, se realizaron capacitaciones mediante videos, tutoriales. En cuanto al diseño del mobiliario, dado que el equipo de desarrollo no se especializa en ese campo, se optó por mantener reuniones con expertos en cirugía (doctores), quienes proporcionaron indicaciones precisas sobre cómo debían ser los objetos dentro del entorno simulado.

- Resultados de pruebas de rendimiento: ver Métricas para medir complejidad de desarrollo, Métricas de Rendimiento en Tiempo Real y Métricas de Experiencia de Usuario (UX).

3.9.5 Fase de Implementación

En la fase de implementación, se capacita a docentes y estudiantes para usar la aplicación, desarrollando manuales y tutoriales interactivos y ofreciendo sesiones de capacitación presenciales o virtuales. La curva de aprendizaje para usar RV puede ser alta para algunos usuarios. Luego, se despliega la aplicación en entornos educativos, distribuyéndola a través del APK de instalación y proporcionando soporte técnico inicial para resolver problemas rápidamente. El despliegue debe ser cuidadoso para evitar problemas técnicos que afecten la experiencia del usuario.

Roles:

- **Coordinador de Capacitación:** Organiza sesiones de formación para docentes y estudiantes (Dr. José Luis Herrera López – Dra. Marianela Mejías de Duarte).
- **Técnico de Soporte:** Resuelve problemas técnicos durante el despliegue (Daniel Sánchez – Kevin Saquinga).
- **Gestor de Proyectos:** Supervisa el lanzamiento y distribución de la aplicación (Dr. José Luis Herrera López – Daniel Sánchez – Kevin Saquinga).
- **Comunicador Técnico:** Elabora manuales y tutoriales para usuarios finales (Daniel Sánchez – Kevin Saquinga).

Entregables:

- Anexo 6: Manual de instalación del sistema y Anexo 7: Manual de Usuario.

3.9.6 Fase de Mantenimiento y Mejora Continua

En la fase de mantenimiento y mejora continua, se monitorea el uso de la aplicación recopilando datos sobre métricas como tiempo de uso, tasas de éxito en simulaciones y retroalimentación de usuarios, implementando herramientas de análisis para rastrear esta información y recopilando comentarios regulares de docentes y estudiantes. El monitoreo continuo puede ser intensivo en recursos. Finalmente, se realizan actualizaciones y mejoras para mantener la aplicación relevante y funcional, incorporando nuevas tecnologías o características según avance el campo de la RV y actualizando el contenido educativo para reflejar avances en la práctica quirúrgica, aunque las actualizaciones deben ser cuidadosamente planificadas para evitar introducir nuevos errores.

Roles:

- **Desarrollador de Actualizaciones:** Implementa mejoras basadas en feedback y Monitorea el uso de la aplicación y recopila métricas (Daniel Sánchez – Kevin Saquinga).
- **Experto Médico:** Valida actualizaciones de contenido médico (Dr. José Luis Herrera López – Dra. Marianela Mejías de Duarte – Dra. Yeisy Cristina Guarate Coronado – Mg. Ana Pamela Pachucho Flores).

Entregables:

- Versiones actualizadas de la aplicación con nuevas características:

En total, se desarrollaron 20 versiones del sistema, cada una registrada mediante commits en la plataforma GitHub. Estas versiones reflejan el proceso iterativo de desarrollo, en el cual se fueron incorporando mejoras, corrigiendo errores y aplicando las sugerencias obtenidas a través de la retroalimentación. El uso de GitHub permitió llevar un control detallado de los avances y facilitar la colaboración entre los miembros del equipo de desarrollo.

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((afb38de...))
$ git checkout d0f93edf78c
Updating files: 100% (414/414), done.
Previous HEAD position was afb38de Finalizacion Del SISTEMA
HEAD is now at d0f93ed Escenario Quirofano
```

Figura 130: Primer commit desarrollado correspondiente a la creación del diseño del escenario

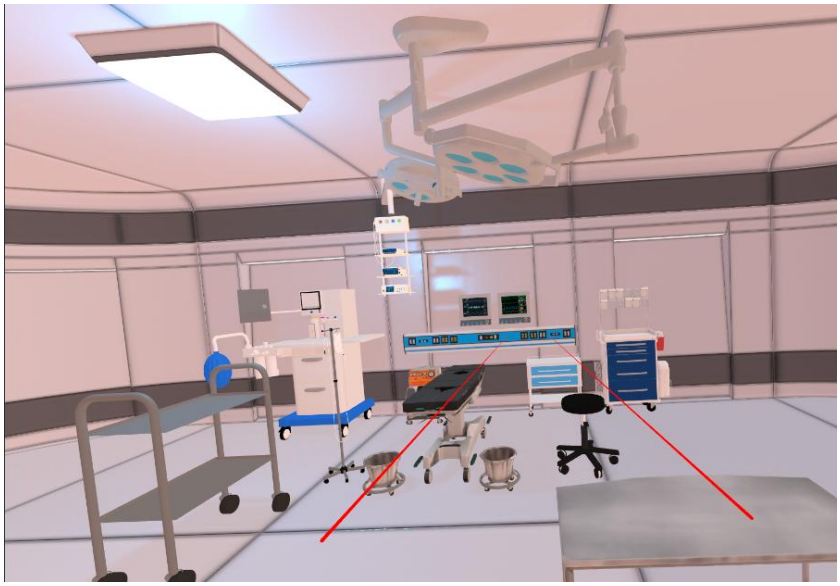


Figura 131:Diseño del quirófano correspondiente al primer commit

Como se observa en la Figura 130, se presenta el primer *commit* desarrollado. Este *commit* se centró en la creación inicial del escenario quirúrgico, estableciendo la base sobre la cual se construiría el resto del entorno simulado. En la Figura 131 se puede observar el resultado final del diseño de dicho escenario, una vez concluido el trabajo correspondiente a esa etapa.

Este avance marcó un hito importante en el desarrollo del sistema, ya que permitió definir la estructura espacial del quirófano y sentó las bases para la integración de funcionalidades posteriores, como la interacción del avatar y la disposición del instrumental quirúrgico.

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((afb38de...))
$ git checkout 25fd66ce17
Previous HEAD position was afb38de Finalizacion Del SISTEMA
HEAD is now at 25fd66c Segundo Commit
```

Figura 132: Segundo commit desarrollado correspondiente a la creación del menú Principal y el menú del instrumental quirúrgico



Figura 133: Menú principal correspondiente al segundo commit

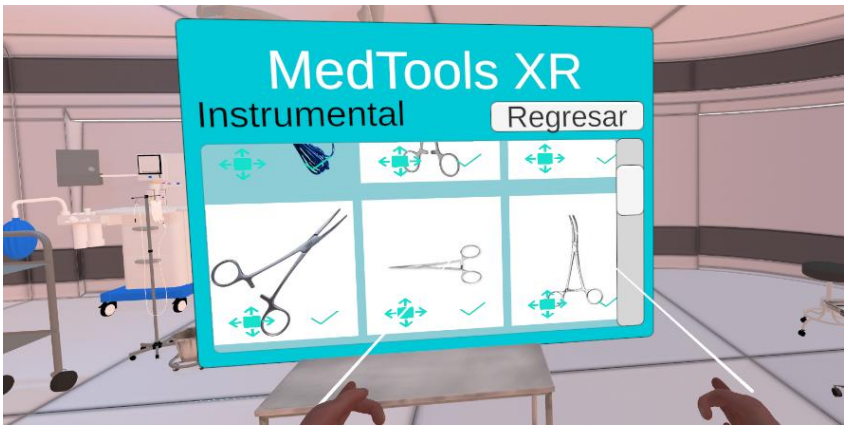


Figura 134: Menú del instrumental quirúrgico correspondiente al segundo commit

Como se observa en la Figura 133 y Figura 134, se presentan los menús creados para la interacción con el usuario. En la Figura 133 se muestra el menú que permite seleccionar el tipo de operación y el tiempo quirúrgico correspondiente. Por otro lado, en la Figura 134 se visualiza el menú que permite al usuario observar los instrumentos quirúrgicos disponibles. Además, en la Figura 132 se muestra el segundo commit realizado como parte del desarrollo de este trabajo.

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((a9196ee...))
$ git checkout a9196ee47c
HEAD is now at a9196ee Descripcion Instrumentos Q.
```

Figura 135: Tercer commit correspondiente a la creación del Canva de expansión para ver información del instrumental seleccionado

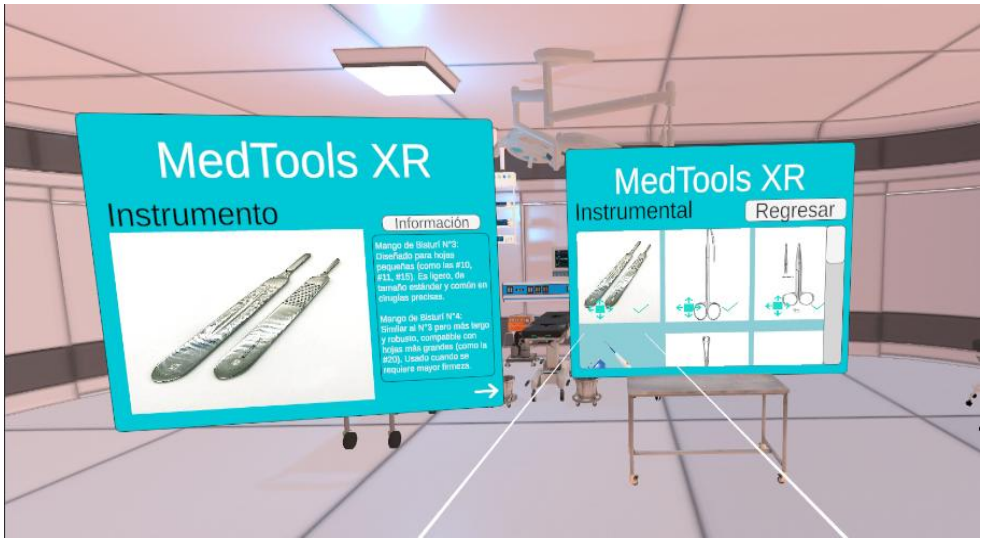


Figura 136: Canva de expansión y detalle de las características principales de la herramienta seleccionada

Como se aprecia en la Figura 135, se muestra el tercer commit desarrollado, el cual se centra en la creación del Canva que permite visualizar en tamaño ampliado del instrumento seleccionado, junto con sus características principales. Esta funcionalidad se puede observar en la Figura 136, donde se presenta un ejemplo de la visualización detallada del instrumental quirúrgico.

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((a9196ee...))
$ git checkout a4d4d589e3
Previous HEAD position was a9196ee Descripcion Instrumentos Q.
HEAD is now at a4d4d58 Cambios
```

Figura 137: Cuarto commit correspondiente a la creación del Canva de expansión para ver información del instrumental seleccionado

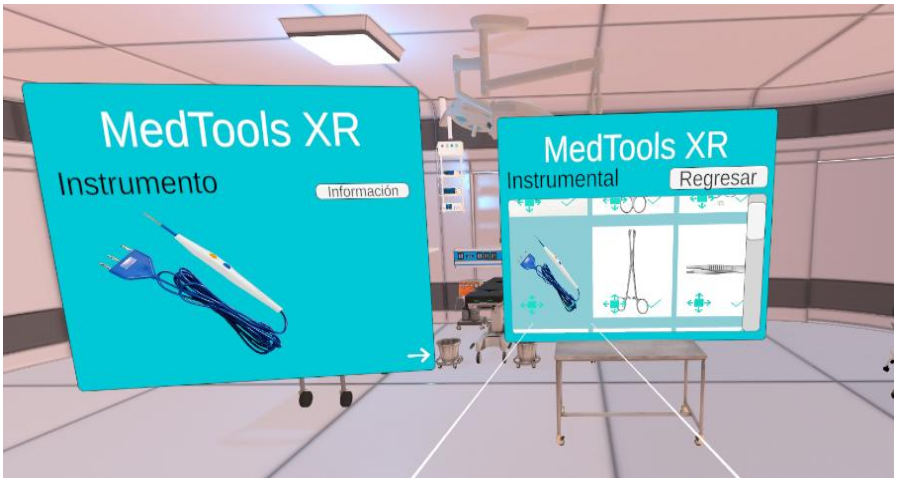


Figura 138: Asignación de todos los botones para dar click aparezca el instrumento en el Canva de expansión

En la Figura 137 se observa el cuarto commit que se desarrolló, el cual se centra en la asignación de funcionalidad a todos los botones, de manera que, al hacer clic sobre un instrumento, este se muestre en el Canvas de expansión. Anteriormente, esta funcionalidad estaba configurada únicamente para un solo instrumento (ver Figura 138). Además, en este commit también se configuró la capacidad del avatar para activar y desactivar los punteros dentro del entorno simulado.

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((a4d4d58...))
$ git checkout 999b6ab6f
Previous HEAD position was a4d4d58 Cambios
HEAD is now at 999b6ab Cambios Coreguidos
```

Figura 139: Quinto commit correspondiente a la configuración de la cámara del avatar



Figura 140: Punto de vista de la cámara a partir del avatar

Como se puede observar en la Figura 140, se configuró el punto de vista de la cámara del avatar, ya que en versiones anteriores la perspectiva se encontraba demasiado baja, lo que obligaba al usuario a realizar movimientos incómodos para tener una vista adecuada. Esta mejora permitió una experiencia más natural y cómoda durante la simulación. Además, en la Figura 139 se muestra commit que de desarrollo para esta tarea.

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((999b6ab...))
$ git checkout 11818e828ac85
Previous HEAD position was 999b6ab Cambios Coreguidos
HEAD is now at 11818e8 Cambios
```

Figura 141: Sexto commit correspondiente a la configuración del avatar

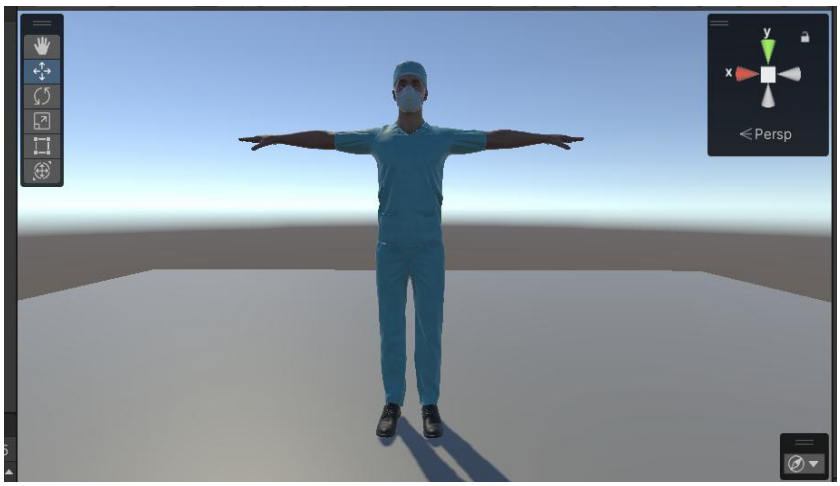


Figura 142: Configuración del avatar para que tenga animaciones

La Figura 141 presenta el sexto commit desarrollado, el cual se centra en la configuración del avatar para incorporar animaciones. Esto se lo puede observar en la Figura 142, donde el avatar en este commit ya puedo movilizarse, anteriormente el avatar se encontraba completamente estático. Con la implementación de las animaciones, ahora el avatar puede simular acciones como caminar, girar, entre otras, lo que mejora considerablemente la experiencia del usuario. En versiones anteriores, el movimiento era rígido y poco natural, lo que afectaba la calidad visual del sistema.

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((11818e8...))
$ git checkout f54d495464
Previous HEAD position was 11818e8 Cambios
HEAD is now at f54d495 Creacion de la mesa Mayo
```

Figura 143:Séptimo commit correspondiente al Canva que simula la mesa mayo

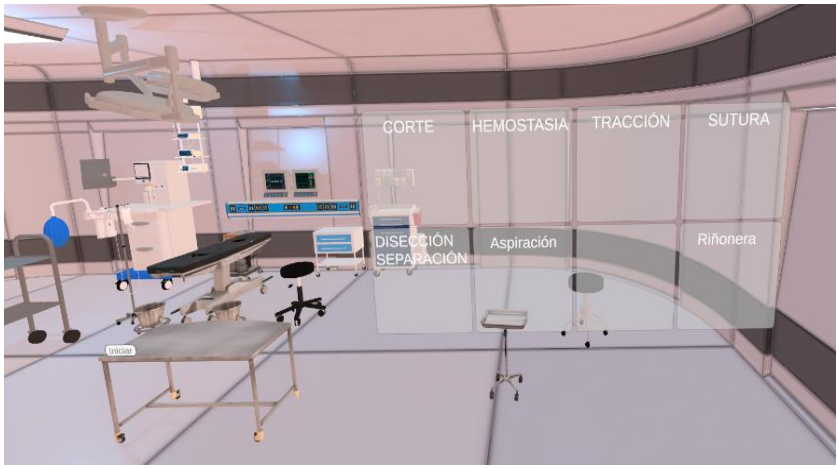


Figura 144: Mesa mayo con sus respectivos tiempos quirúrgicos

Como se ilustra en la Figura 143, se presenta el séptimo commit realizado en el sistema, el cual consiste en la creación del Canva que simula la mesa Mayo. En la Figura 144 se puede observar cómo quedo el diseño del Canva de la mesa mayo. Dado que se están utilizando imágenes para representar los instrumentales, se optó por esta solución, ya que resulta más efectivo y visualmente claro tener el Canva ubicado frente al usuario dentro del entorno de simulación.

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((f54d495...))
$ git checkout 5639eb6e7cf62073
Previous HEAD position was f54d495 Creacion de la mesa Mayo
HEAD is now at 5639eb6 Cambios Mesa Mayo
```

Figura 145: Octavo commit correspondiente a cambios de la mesa mayo

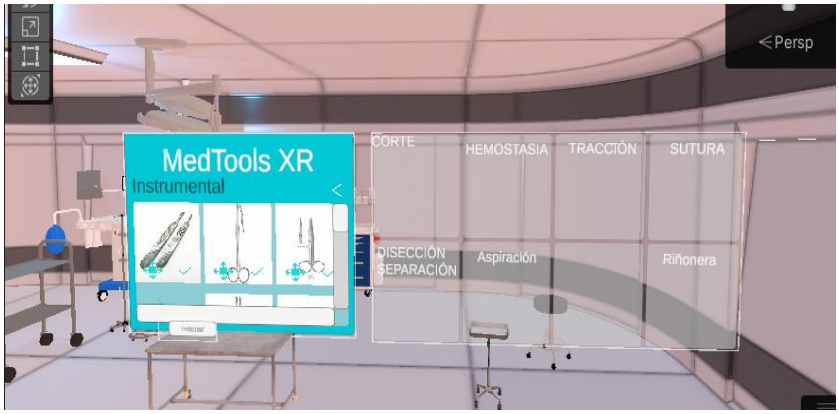


Figura 146: Mesa mayo junto al menú del instrumental quirúrgico

Según se observa en la Figura 146, se realizó la modificación para que el menú del instrumental quirúrgico aparezca junto al Canvas de la mesa Mayo. En versiones anteriores, al ejecutar uno de los elementos desaparecía, lo que dificultaba la interacción del usuario. Esta mejora fue registrada en el commit (ver Figura 145).

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((5639eb6...))
$ git checkout c5e4a2b5259
Previous HEAD position was 5639eb6 Cambios Mesa Mayo
HEAD is now at c5e4a2b COntrol del canvas Mesa Mayor - error de aviso
```

Figura 147: Noveno commit correspondiente a la creación de un aviso si se comete errores



Figura 148: Mensaje de error al seleccionar mal un instrumento

Como puede verse en la Figura 148, se implementó un mensaje de advertencia que se ejecuta cada vez que el usuario selecciona un instrumento que no corresponde al tiempo quirúrgico actual. Esta funcionalidad permite guiar al usuario y evitar errores

durante el proceso de armado de la mesa Mayo. Esta mejora fue registrada en el commit (ver Figura 147).

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((c5e4a2b...))
$ git checkout a36af67e488
Updating files: 100% (94/94), done.
Previous HEAD position was c5e4a2b COntrol del canvas Mesa Mayor - error de avis
o
HEAD is now at a36af67 Configuracion de los botones Mesa Mayo
```

Figura 149: Décimo commit correspondiente a la configuración de los botones para la interacción con la mesa mayo



Figura 150: Asignación del instrumental quirúrgico en la mesa mayo

Tal como se muestra en la Figura 149, se puede observar el décimo commit, el cual corresponde a la configuración de los botones para que, al hacer clic, el instrumental seleccionado se posicione automáticamente en el panel correspondiente. Esta funcionalidad se visualiza en la Figura 150, donde se evidencia el comportamiento esperado dentro del entorno del simulador.

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((a36af67...))
$ git checkout d5c5d9cb328
Previous HEAD position was a36af67 Configuracion de los botones Mesa Mayo
HEAD is now at d5c5d9c Cambios toggle
```

Figura 151: Onceavo commit correspondiente al Toggle para activar la mesa mayo

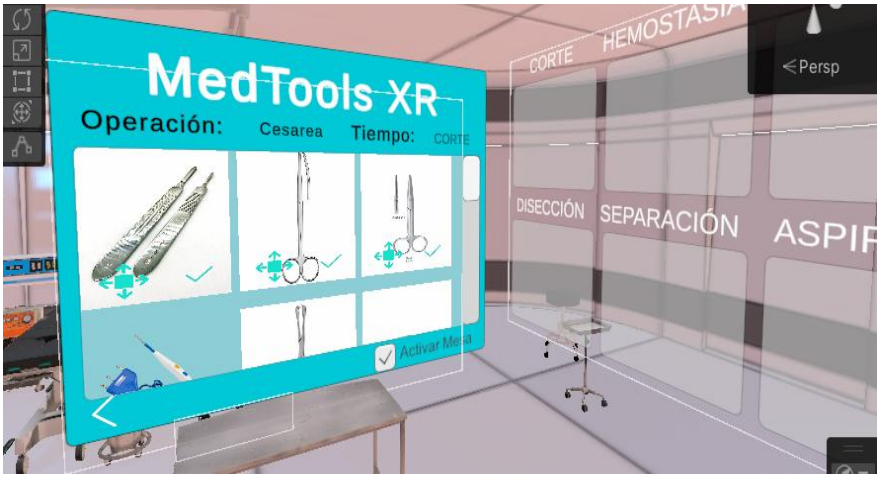


Figura 152: Configuración del Toggle al dar click se activa el Canva de la mesa mayo

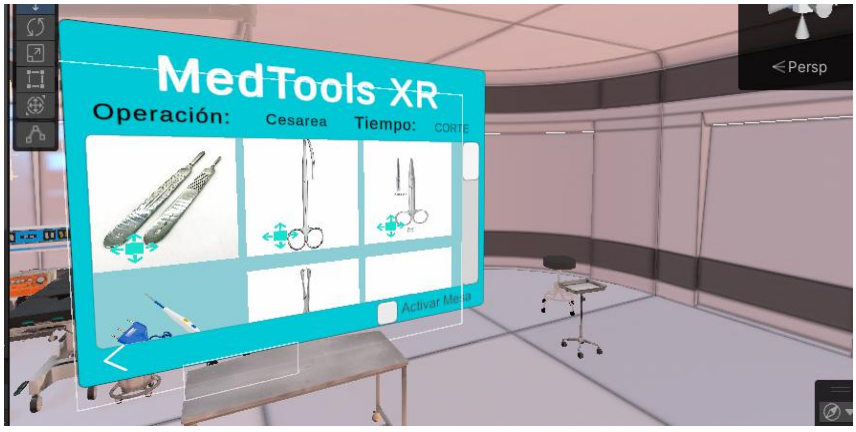


Figura 153: Configuración del Toggle al dar volver a click se desactiva el Canva de la mesa mayo

Como se muestra en la Figura 152, al activar el toggle denominado "Activar mesa Mayo", el Canva correspondiente a la mesa Mayo se activa automáticamente. Si se vuelve a hacer clic en el mismo toggle, el Canva se desactiva, tal como se observa en la Figura 153. Esta funcionalidad fue implementada y registrada en el onceavo commit (ver Figura 151).

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((d5c5d9c...))
$ git checkout b11cc6542d92c1
Previous HEAD position was d5c5d9c Cambios toggle
HEAD is now at b11cc65 Control de camara del avatar solucionado
```

Figura 154: Doceavo commit correspondiente al control de la cámara del avatar

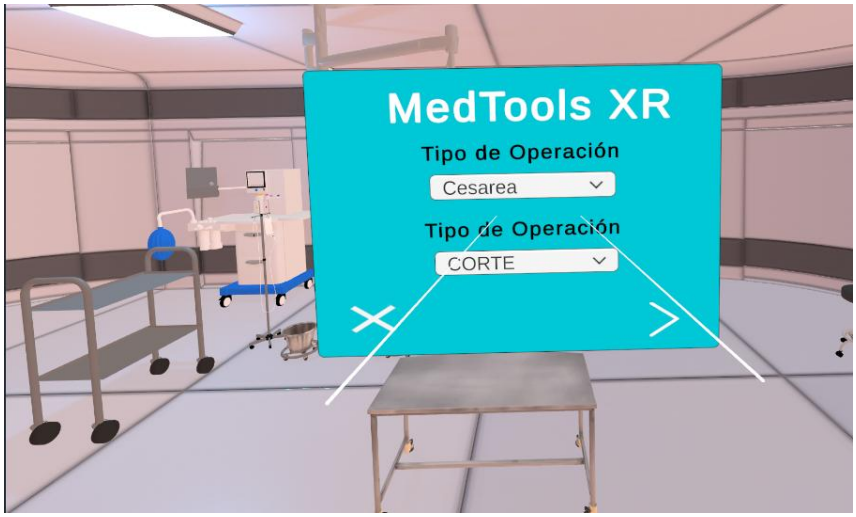


Figura 155: Visión de la cámara integrada con el avatar

En la Figura 154 se observa el doceavo commit, el cual se enfoca en la configuración de la cama en relación con el avatar. Aunque este aspecto ya había sido corregido previamente, durante las pruebas con los usuarios se detectó que la visión de la cámara variaba dependiendo del lugar físico donde se ejecutaba la simulación (ver Figura 155). Este problema fue solucionado configurando una vista de cámara global, que se mantiene estable sin importar el entorno físico en el que se realicen las pruebas.

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((b11cc65...))
$ git checkout 2ad3deffd74c7f1
Previous HEAD position was b11cc65 Control de camara del avatar solucionado
HEAD is now at 2ad3def Cambios de objetos mesa y mesa mayo
```

Figura 156: Treceavo commit correspondiente a los cambios de los muebles del escenario



Figura 157: Control para la activación de los objetos (mesa semi luna y de la mesa mayo)

En la Figura 157 se puede observar cómo los objetos, tanto la mesa semicircular como la mesa Mayo, fueron modificados. Este cambio se realizó debido a las observaciones recibidas durante las revisiones, en las cuales se especificó que, en la práctica real, el procedimiento correcto consiste en cubrir o vestir dichos objetos antes de iniciar con el armado de la mesa Mayo. Esta mejora fue registrada en el commit (ver Figura 156).

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((2ad3def...))
$ git checkout fc02bf2f04d0
Previous HEAD position was 2ad3def Cambios de objetos mesa y mesa mayo
HEAD is now at fc02bf2 Creacion del menu General
```

Figura 158: Catorceavo commit correspondiente a los canvas que nos permiten evaluar el armado de la mesa mayo



Figura 159: Menú principal más el botón General funcional

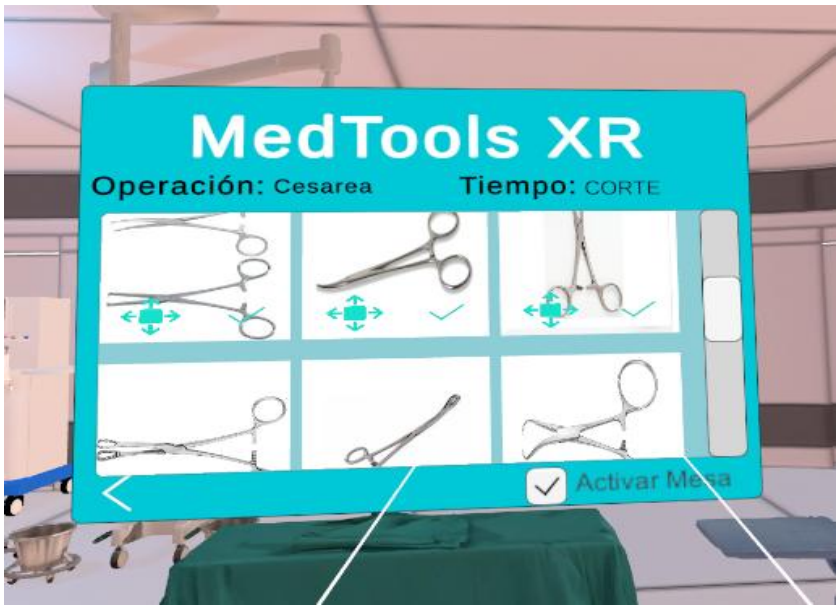


Figura 160: Menú de los instrumentos para la ver el aprendizaje

Como se aprecia en la Figura 159, se muestra el menú que permite seleccionar el tipo de operación y el tiempo quirúrgico. Además, incluye el botón "General", cuya función es tomar únicamente el valor de la operación seleccionada. Una vez que se hace clic en este botón, se abre el menú de los instrumentos, como se puede ver en la Figura 160. A diferencia de versiones anteriores, en las que solo se permitía escoger los instrumentos correspondientes al tiempo quirúrgico seleccionado, esta nueva funcionalidad permite que, una vez seleccionados todos los instrumentos del primer tiempo quirúrgico, el sistema avance automáticamente al siguiente tiempo, y así

sucesivamente, hasta completar el armado de toda la mesa Mayo, todo este proceso se lo guardo en el commit catorceavo (ver Figura 158).

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((fc02bf2...))
$ git checkout cde426318886
Previous HEAD position was fc02bf2 Creacion del menu General
HEAD is now at cde4263 Correcciones del menu y creacion del nombre flotante de l
os objetos
```

Figura 161: Quinceavo commit centrado en la creación de los nombres flotantes



Figura 162: Creación de nombre flotante al pasar los punteros

Según se observa en la Figura 162, se implementó un nombre flotante que aparece cuando el usuario pasa el puntero sobre un objeto, mostrando así su nombre correspondiente. Esta funcionalidad mejora la identificación de los instrumentos dentro del entorno de simulación. El desarrollo de esta característica fue registrado en el quinceavo commit (ver Figura 161).

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((cde4263...))
$ git checkout f6cadf011e1
Previous HEAD position was cde4263 Correcciones del menu y creacion del nombre f
lotante de los objetos
HEAD is now at f6cadf0 Creacion del menu principal y del tutorial
```

Figura 163: Décimo sexto commit enfocado en la creación del menú de bienvenida con sus tutoriales



Figura 164: Menú de bienvenida al sistema de realidad virtual



Figura 165: Tutorial del uso de los joisntink

En la Figura 163 se presenta el décimo sexto commit, el cual se centra en la creación del menú de bienvenida que se muestra al ejecutar el sistema (ver Figura 164). Dentro de este menú también se incluyen los tutoriales, con el objetivo de orientar al usuario sobre el funcionamiento y la forma de interactuar correctamente una vez que se encuentre dentro del escenario del quirófano (ver Figura 165).

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((f6cadf0...))
$ git checkout 74532e9454cc6
Previous HEAD position was f6cadf0 Creacion del menu principal y del tutorial
HEAD is now at 74532e9 Toques Finales
```

Figura 166: Décimo séptimo commit enfocado en la asignación del resto de los nombres flotantes a los objetos



Figura 167: Nombres flotantes asignados al resto de los objetos

De acuerdo con Figura 167, se observa que, con fines educativos, se realizó un cambio en el diseño de los nombres flotantes. El nuevo diseño permite que la información se entienda de una manera más clara y didáctica. Además, esta mejora se aplicó a todos los objetos del entorno, ya que en el commit anterior solo se había implementado para un único objeto. Esta actualización fue registrada en el commit décimo séptimo (ver Figura 166).

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((74532e9...))
$ git checkout 10cf0ea2c1
Previous HEAD position was 74532e9 Toques Finales
HEAD is now at 10cf0ea Arreglo de la pantalla congelada
```

Figura 168: Décimo octavo commit centrado en la corrección de la pantalla congelada



Figura 169: Canva de Cargando … para controlar la pantalla congelada

En la Figura 169, se puede observar el Canva que fue creado con un mensaje que indica "Cargando...", acompañado de una slider. Esta slider se va llenando progresivamente a medida que se cargan todos los componentes del escenario del quirófano. Todo este proceso se realiza de manera sincronizada, de modo que, una vez que el slider se completa, el sistema cambia automáticamente al entorno del quirófano. Con esta implementación se solucionó el problema del congelamiento de la pantalla durante la carga. Esta mejora fue registrada en el commit décimo octavo (ver Figura 168).

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((10cf0ea...))
$ git checkout 3e9e3c66be91
Previous HEAD position was 10cf0ea Arreglo de la pantalla congelada
HEAD is now at 3e9e3c6 Cambios finales
```

Figura 170: Décimo noveno commit centrado en el Canva que te permite ver los resultados del armado de la mesa mayo

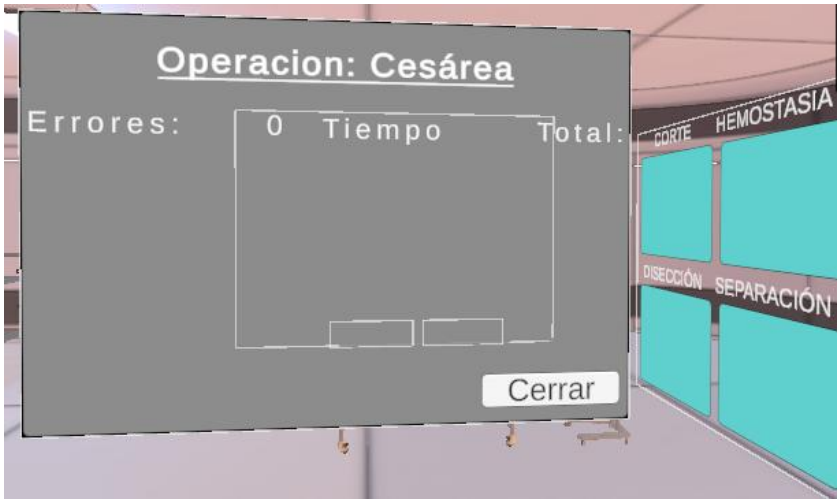


Figura 171: Canva para ver los errores cometidos en el armado

De acuerdo con la Figura 170, se presenta el penúltimo commit, el cual se enfoca en la creación de un Canva que permite visualizar los errores cometidos en cada tiempo quirúrgico. Además, este panel muestra el tiempo total que el usuario tardó en armar la mesa Mayo, brindando así una retroalimentación completa del desempeño. Esta funcionalidad (ver Figura 171).

```
User@kevin MINGW64 ~/Downloads/Proyecto/Proyecto ((3e9e3c6...))
$ git checkout afb38def8f37f65f
Previous HEAD position was 3e9e3c6 Cambios finales
HEAD is now at afb38de Finalizacion Del SISTEMA
```

Figura 172: Ultimo commit desarrollado para el sistema



Figura 173: Canva de notificación de finalizar todos los tiempos

En la Figura 172 se puede observar el último commit desarrollado, el cual está centrado en la creación de un Canva que notifica al usuario cuando ha finalizado correctamente todos los tiempos quirúrgicos. Esta notificación proporciona un cierre claro del proceso de armado de la mesa Mayo (ver Figura 173).



Figura 174: Menú de tipo de operación corregido



Figura 175: Menú de tiempo de operación corregido

Además de lo anterior, se realizaron las correcciones correspondientes a las faltas de ortografía, ya que en los campos relacionados con los tiempos quirúrgicos y los tipos de operación no se manejaban adecuadamente las tildes. Inicialmente, esto generaba problemas en la programación, pues las palabras con tilde no eran reconocidas correctamente por el sistema. Para solucionar este inconveniente, se implementó un método de normalización que permitió ignorar las diferencias entre palabras acentuadas y no acentuadas, así como las variaciones entre mayúsculas y minúsculas. Gracias a este ajuste, se logró una mayor consistencia en el procesamiento de datos, mejorando la funcionalidad general del sistema (ver Figura 174, Figura 175) .

3.10 Validación de la metodología

Para validar la metodología empleada en el desarrollo del sistema, fue necesario considerar un enfoque que permita no solo la creación del producto, sino también su evaluación continua y mejora iterativa. En este contexto, se recurrió a un modelo metodológico que integra tanto el diseño como la investigación aplicada en entornos tecnológicos. A continuación, se presenta la metodología Design Science Research (DSR), seleccionada por su enfoque orientado a la construcción y validación de artefactos innovadores.

3.10.1 Design Science Research (DSR) – Investigación Basada en Diseño

El Design Science Research (DSR) es una metodología de investigación contemporánea usada ampliamente en Ingeniería de Software, Sistemas de Información y Tecnologías Emergentes (como la RV). Se enfoca en la creación, evaluación y mejora de artefactos tecnológicos, como tu metodología de desarrollo de software [108].

3.10.2 Etapas de Design Science Research

Tabla 102: Etapas que existen Design Science Research (DSR) – Investigación Basada en Diseño

Etapa del DSR	**Aplicación a tu metodología RV para enfermería/cirugía**
Identificación del problema	Analizar brechas actuales en formación clínica con RV.
Definición de objetivos	Establecer metas claras de la metodología (ej. realismo, usabilidad, aprendizaje).
Diseño y desarrollo del artefacto	Desarrollar tu metodología como modelo formal con fases, roles, entregables.
Demostración	Aplicar la metodología en un proyecto piloto de desarrollo RV en la facultad de enfermería.
Evaluación del artefacto	Usar técnicas mixtas: métricas de desempeño, encuestas TAM, feedback clínico.
Comunicación	Publicar resultados en papers, congresos, tesis u observatorios de innovación educativa.

La metodología Design Science Research (DSR) se estructura en una serie de etapas que permiten guiar de forma sistemática el desarrollo y validación de artefactos tecnológicos. En la siguiente Tabla 102 se presentan dichas etapas y su aplicación concreta en la metodología propuesta basada en realidad virtual (RV) para el área de

enfermería y cirugía, destacando cómo cada fase contribuye a la creación de un entorno educativo efectivo y validado en el ámbito clínico.

a. Identificación del Problema

Objetivo: Confirmar la necesidad de una metodología formal para desarrollo de software RV en salud.

Actividades:

- Aplicar entrevistas semiestructuradas a médicos, ingenieros sobre limitaciones actuales en simulación clínica.
- Encuesta diagnóstica a los 70 estudiantes para identificar dificultades en el aprendizaje práctico.
- Revisión documental de proyectos previos de simulación sin metodología estructurada.

Instrumentos:

- Guía de entrevista ver Anexo 2: Entrevistas
- Cuestionario diagnóstico en Google Forms ver Anexo 5: Metodo de evaluación TAM

b. Definición de Objetivos de la Solución

Objetivo: Establecer criterios de éxito para la metodología (ej. realismo, usabilidad, precisión técnica, aprendizaje).

Actividades:

- Taller colaborativo con desarrolladores y médicos para consensuar atributos deseables.
- Focus group con estudiantes seleccionados (n=12) para retroalimentación inicial.

Instrumentos:

Registro audiovisual del taller visitar el enlace: https://drive.google.com/drive/folders/14B2rvV9vI-eDQq6Jqqa-Zd1zjAhAKIMr?usp=sharing

Como se puede observar en la Figura 176, Figura 177 y Figura 178, se presentan las capacitaciones recibidas para comprender el procedimiento de armado de la mesa mayo.



Figura 176: Capacitación del armado de la mesa mayo



Figura 177: Capacitación de un quirófano funcional en el HGDA



Figura 178: Capacitación de los instrumentos quirúrgicos

c. Diseño y Desarrollo del Artefacto (Metodología)

Objetivo: Construir formalmente la metodología en fases, roles y entregables.

Actividades:

- Documento de especificación de la metodología.
- Mapeo de roles: médicos expertos; desarrolladores técnicos; estudiantes validadores.
- Desarrollo conjunto de una guía práctica.

Instrumentos:

Plantilla ISO 29148 para especificación ver Especificación de Requerimientos de Software (SRS) basado en la norma ISO 29148:2018 ISO 29148

Herramientas colaborativas: Se utilizaron diversas herramientas colaborativas para el desarrollo del proyecto. Para el control de versiones se empleó GitHub; en el diseño de prototipos, se utilizó Figma. En cuanto a los lenguajes de programación, se trabajó con C#, por ser uno de los más utilizados en el desarrollo de aplicaciones de realidad virtual. Para el motor de videojuegos se optó por Unity, debido a su amplio soporte por parte de la comunidad. Finalmente, para el diseño de los muebles se empleó Blender.

d. Demostración

Objetivo: Aplicar la metodología en un caso real de desarrollo: simulador de identificación de instrumentos quirúrgicos en RV.

Actividades:

Los 2 desarrolladores siguen la metodología para construir el escenario.

Los médicos proporcionan retroalimentación clínica en diseño y pruebas.

Se involucran 70 estudiantes usando el simulador final en sesiones prácticas.

Instrumentos:

- Bitácora del proyecto ver Versiones actualizadas de la aplicación con nuevas características donde se detalla los commits que se lo realizaron en todo el proyecto

- Captura de sesiones VR:



Figura 179: Demostración práctica del entorno de realidad virtual



Figura 180: Práctica guiada en la navegación del entorno simulado



Figura 181: Instrucciones para la interacción dentro del simulador de realidad virtual.



Figura 182: Capacitación en la manipulación de instrumentos virtuales en la mesa mayo

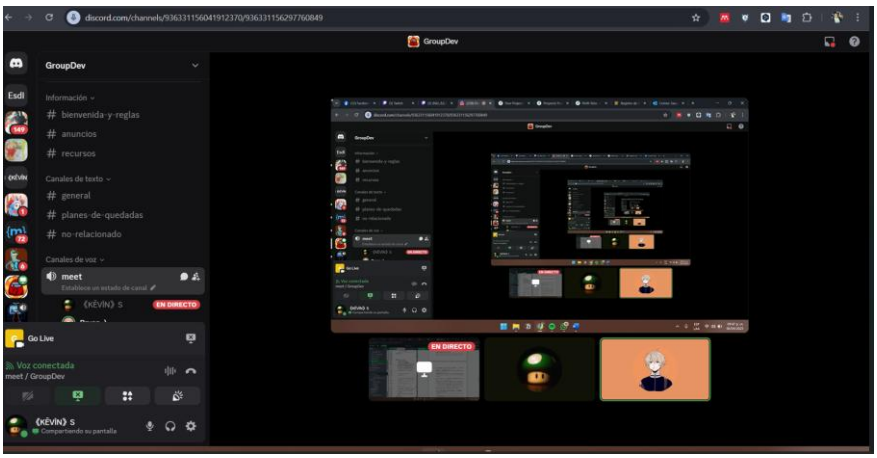


Figura 183: Reunión entre desarrolladores para discutir avances del sistema.

Como se pueden observar en la Figura 179, Figura 180, Figura 181, Figura 182 y Figura 183 , se presentan las capacitaciones realizadas a los estudiantes que utilizarán el sistema.

e. Evaluación

Objetivo: Validar el rendimiento, usabilidad y aprendizaje generado por la metodología.

Actividades por Grupo:

Tabla 103: Actividades a ejecutar para evaluar la metodología

Grupo	**Actividades**	**Instrumentos**
Médicos	Evaluación de realismo clínico, precisión de procesos, seguridad de simulación	Rúbrica de validación clínica (Likert)
Desarrolladores	Evaluación de claridad, escalabilidad y utilidad de la metodología	Cuestionario TAM + entrevistas
Estudiantes	Pruebas prácticas con simulador + encuesta de satisfacción	SUS, NASA-TLX (Índice de carga)

f. Comunicación

Objetivo: Sistematizar los resultados y compartirlos académica o profesionalmente.

Actividades:

- Elaboración de informe técnico completo (TESIS).

Instrumentos:

- Informe PDF estructurado

3.10.3 Resultados de la encuesta TAM

Con el propósito de validar una metodología orientada al desarrollo de aplicaciones de realidad virtual, se aplicó una encuesta estructurada a un grupo de ocho profesionales con experiencia en el área de Tecnologías de la Información. La encuesta (ver Anexo

5: Metodo de evaluación TAM)fue diseñada para recoger percepciones en torno a aspectos clave del proceso metodológico, tales como su utilidad, facilidad de uso, actitud hacia su aplicación y la intención de uso futuro. Aunque el enfoque de la metodología tiene aplicación en contextos clínico-educativos, la validación se centró en su estructura, claridad y aplicabilidad desde una perspectiva técnica y de desarrollo. A continuación, se presentan los resultados obtenidos, los cuales reflejan la valoración de estos expertos sobre la solidez y viabilidad de la propuesta metodológica.

Tabla 104: Promedio y Media de la Utilidad Percibida de la Metodología

Utilidad Percibida (UP)		
	PROMEDIO POR PREGUNTA	**MEDIA POR PREGUNTA**
UP 1	4.75	5
UP2	4.75	5
UP3	4.75	5
UP4	5	5
UP5	4.75	5
PROMEDIO GENERAL	4.8	
MEDIA GENERAL	5	

La Sección 1: Utilidad Percibida (ver Tabla 104) evaluó la percepción de utilidad de la metodología aplicada en el contexto clínico-educativo. Los resultados reflejan una valoración altamente positiva por parte de los participantes.

- Cada uno de los ítems evaluados (UP1 a UP5) obtuvo un promedio por pregunta entre 4.75 y 5.00, lo que indica un nivel de acuerdo muy alto con las afirmaciones planteadas.

- La media por pregunta fue uniforme en todos los casos, con un valor de 5, evidenciando que la respuesta más frecuente para cada afirmación fue "Totalmente de acuerdo".

En conjunto:

- El promedio general de la sección fue de 4.8, lo cual sugiere una percepción muy favorable de la metodología en cuanto a su utilidad percibida.
- La media general fue de 5, consolidando el alto grado de consenso entre los participantes respecto a los beneficios clínicos y educativos de la metodología.

Tabla 105: Promedio y Media de la Facilidad de Uso Percibida de la Metodología

Facilidad de Uso Percibida (FUP)		
	PROMEDIO POR PREGUNTA	**MEDIA POR PREGUNTA**
FUP1	4.5	5
FUP2	4.75	5
FUP3	4.25	4.5
FUP4	4.25	4
FUP5	4.5	4.5
PROMEDIO GENERAL	4.45	
MEDIA GENERAL	5	

Como se observa en la Tabla 105 la Sección 2: Facilidad de Uso Percibida (FUP) se evaluó la facilidad con la que los distintos actores comprenden y aplican la metodología propuesta para el desarrollo de aplicaciones de realidad virtual en contextos clínicos.

- En cuanto al promedio por pregunta, los valores oscilaron entre 4.25 y 4.75, lo cual indica un nivel alto de acuerdo respecto a la comprensión, claridad de roles y facilidad de implementación de la metodología.
- Las medias por pregunta se ubicaron entre 4.0 y 5.0, reflejando que la respuesta más frecuente fue mayormente “Totalmente de acuerdo”, aunque con ligeras variaciones que evidencian matices en la percepción de algunos ítems específicos.

Además:

- El promedio general de la sección fue de 4.45, confirmando una percepción globalmente positiva respecto a la facilidad de uso de la metodología.
- La media general fue de 5, lo que indica que el valor más frecuente fue nuevamente el más alto de la escala, consolidando así una tendencia de alta aceptación.

Tabla 106: Promedio y Media de la Actitud hacia el Uso de la Metodología

Actitud hacia el uso (ATU)		
	PROMEDIO POR PREGUNTA	**MEDIA POR PREGUNTA**
ATU1	4.25	4.5
ATU2	5	5
ATU3	4.75	5
PROMEDIO GENERAL	4.67	
MEDIA GENERAL	5	

La Sección 3: Actitud hacia el uso (ATU) se evaluó la actitud general de los usuarios hacia la metodología propuesta, considerando tanto su disposición emocional como racional frente a su uso en contextos educativos clínicos (ver Tabla 106).

- Los promedios por pregunta se situaron entre 4.25 y 5.00, lo cual indica una actitud predominantemente positiva hacia la metodología.
- Las medias por pregunta oscilaron entre 4.5 y 5.0, confirmando que la mayoría de los participantes eligieron niveles altos de acuerdo, con especial énfasis en la aceptación de su aplicación en educación clínica y su deseo de adopción en proyectos similares.

En conjunto:

- El promedio general de esta sección fue de 4.67, lo cual muestra una actitud fuertemente favorable hacia el uso de la metodología.

- La media general fue de 5, evidenciando que la respuesta más común fue “Totalmente de acuerdo”.

Tabla 107: Promedio y Media de la Intención de Uso de la Metodología

Intención de Uso (IU)		
	PROMEDIO POR PREGUNTA	**MEDIA POR PREGUNTA**
IU1	4.75	5
IU2	4.75	5
IU3	4.75	5
PROMEDIO GENERAL	4.75	
MEDIA GENERAL	5	

La Sección 4: Intención de Uso (IU) (ver Tabla 107) midió la probabilidad de adopción futura de la metodología por parte de los usuarios, considerando su disposición a recomendarla, reutilizarla y adoptarla como estándar en el desarrollo de simulaciones clínicas.

- Los promedios por pregunta fueron consistentes, con un valor de 4.75 en los tres ítems, lo cual indica una valoración muy alta de la intención de uso.
- Las medias por pregunta fueron todas de 5, evidenciando que la mayoría de los encuestados optó por la opción "Totalmente de acuerdo".

Además:

- El promedio general de la sección fue de 4.75, lo cual confirma una intención sólida de continuar utilizando la metodología.
- La media general fue de 5, reflejando una fuerte tendencia a recomendar y adoptar este enfoque metodológico.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La realidad virtual se posiciona como una alternativa eficaz ante las limitaciones de infraestructura física en la formación en instrumentación quirúrgica. A través de un análisis técnico, se seleccionaron herramientas como Unity, C#, Meta Quest 3 y Blender por su compatibilidad y accesibilidad. Las encuestas aplicadas a estudiantes evidenciaron una alta aceptación de esta tecnología, destacando su utilidad para identificar instrumentos, desarrollar habilidades clínicas y fomentar la práctica autónoma. El levantamiento de requerimientos permitió definir las necesidades técnicas y pedagógicas del sistema, estableciendo una base sólida para su desarrollo funcional.

- Para evaluar la complejidad del software de simulación en realidad virtual, se aplicaron métricas técnicas antes de usar una metodología planteada. Inicialmente, se analizaron aspectos estructurales del código como la Cyclomatic Complexity , Halstead Complexity. Luego, se midió el rendimiento en tiempo real, considerando tiempos de respuesta y uso de recursos. También se evaluó la experiencia del usuario con escalas como SUS y NASA-TLX. Con esta información, se aplicó la metodología propuesta, ajustando la interfaz y optimizando funciones con base en datos técnicos y percepciones reales de los usuarios.

- Se logró implementar exitosamente un sistema de realidad virtual inmersiva para simular la instrumentación quirúrgica en cirugías abiertas, enfocado en la formación de estudiantes de enfermería. El desarrollo se basó en la metodología Extreme Programming (XP), permitiendo un proceso iterativo y flexible. Se incorporaron funciones clave como navegación, armado de la mesa Mayo, selección de tiempos quirúrgicos y retroalimentación interactiva. El sistema fue optimizado para dispositivos Meta Quest 3, asegurando compatibilidad, rendimiento y usabilidad. Las pruebas de aceptación confirmaron su efectividad como recurso educativo realista y funcional.

4.2 Recomendaciones

- Se recomienda que futuras investigaciones amplíen la variedad de tecnologías evaluadas en el ámbito de la simulación quirúrgica con realidad virtual, considerando no solo distintos motores gráficos y dispositivos de visualización, sino también tecnologías emergentes como la realidad mixta y la inteligencia artificial. Esto permitiría enriquecer las posibilidades de simulación, mejorar la personalización del aprendizaje y adaptar los entornos virtuales a distintos contextos clínicos y pedagógicos.

- A partir de las métricas aplicadas, se recomienda implementar un monitoreo continuo de calidad que integre indicadores técnicos y evaluaciones de la experiencia del usuario. Esto facilitará la detección de fallos, la mejora del rendimiento y la sostenibilidad del sistema. La metodología utilizada en este estudio también podría aplicarse en otros proyectos de realidad virtual, permitiendo estructurar de forma eficiente las etapas de desarrollo y evaluación. Su uso en distintos contextos serviría para comprobar su efectividad y adaptar el modelo según las necesidades de otras áreas formativas.

- Finalmente, se recomienda que en fases futuras del proyecto se integre un módulo de gestión de usuarios con respaldo en base de datos, lo que permitiría registrar el progreso individual de cada estudiante y reanudar las prácticas desde el punto de interrupción. También se sugiere la inclusión de instrumental quirúrgico modelado en 3D para aumentar el realismo y valor didáctico de la simulación. Cabe señalar que el sistema actual está diseñado específicamente para dispositivos Meta Quest 2, Quest 3 y Quest 3S, y fue desarrollado con Unity en su versión 2022.3.3.18f1. Se recomienda su implementación en instituciones educativas del área de la salud como una herramienta complementaria para la enseñanza práctica en entornos controlados.

Referencias Bibliográficas

[1] X. Li, D. Elnagar, G. Song, and R. Ghannam, "Advancing Medical Education Using Virtual and Augmented Reality in Low- and Middle-Income Countries: A Systematic and Critical Review," *Virtual Worlds*, vol. 3, no. 3, pp. 384–403, Sep. 2024, doi: 10.3390/virtualworlds3030021.

[2] J. Einloft *et al.*, "Immersive medical training: a comprehensive longitudinal study of extended reality in emergency scenarios for large student groups," *BMC Med Educ*, vol. 24, no. 1, Dec. 2024, doi: 10.1186/s12909-024-05957-3.

[3] M. Mergen, N. Graf, and M. Meyerheim, "Reviewing the current state of virtual reality integration in medical education - a scoping review," *BMC Med Educ*, vol. 24, no. 1, Jul. 2024, doi: 10.1186/s12909-024-05777-5.

[4] S. Pedram, R. Ogie, S. Palmisano, M. Farrelly, and P. Perez, "Cost–benefit analysis of virtual reality-based training for emergency rescue workers: a socio-technical systems approach," *Virtual Real*, vol. 25, no. 4, pp. 1071–1086, Dec. 2021, doi: 10.1007/S10055-021-00514-5/METRICS.

[5] M. M. Khan, M. N. H. Bhuiyan, and T. A. Tania, "Research and Development of Virtual Reality Application for Teaching Medical Students," *2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021*, 2021, doi: 10.1109/ICCCNT51525.2021.9580004.

[6] R. Sultanova and R. Sharaeva, "Virtual reality-based immersive simulation mechanics for invasive surgery training," *Proceedings - International Conference on Developments in eSystems Engineering, DeSE*, vol. October-2019, pp. 924–928, Oct. 2019, doi: 10.1109/DESE.2019.00171.

[7] M. Queisner, M. Pogorzhelskiy, C. Remde, J. Pratschke, and I. M. Sauer, "VolumetricOR: A New Approach to Simulate Surgical Interventions in

Virtual Reality for Training and Education," *Surg Innov*, vol. 29, no. 3, pp. 406–415, Jun. 2022, doi: 10.1177/15533506211054240.

[8] N. Ashtari, A. Bunt, J. McGrenere, M. Nebeling, and P. K. Chilana, "Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities," *Conference on Human Factors in Computing Systems - Proceedings*, Apr. 2020, doi: 10.1145/3313831.3376722/SUPPL_FILE/A593-ASHTARI-PRESENTATION.MP4.

[9] M. D. Negrão, W. Ferreira, B. Bohrer, C. M. D. S. Freitas, A. Maciel, and L. Nedel, "Design and Think-Aloud Study of an Immersive Interface for Training Healthcare Professionals in Clinical Skills," *ACM International Conference Proceeding Series*, pp. 157–165, Nov. 2023, doi: 10.1145/3625008.3625037/SUPPL_FILE/SVR.MP4.

[10] K. Hoek, C. Jaschinski, M. van Velzen, and E. Sarton, "Development of a Real-Time Adaptable Virtual Reality-Scenario Training for Anaesthesiology Education, a User-Centered Design," *International Conference on Computer Supported Education, CSEDU - Proceedings*, vol. 1, pp. 751–757, 2024, doi: 10.5220/0012755600003693.

[11] L. Steffny *et al.*, "Design, Development, and Evaluation of a Virtual Reality-Based Distance Learning Application in Manual Medicine and Therapy," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14708 LNCS, pp. 154–169, 2024, doi: 10.1007/978-3-031-61047-9_10.

[12] M. la Cour, A. S. S. Thomsen, M. Alberti, and L. Konge, "Simulators in the training of surgeons: is it worth the investment in money and time? 2018 Jules Gonin lecture of the Retina Research Foundation," *Graefes Arch Clin Exp Ophthalmol*, vol. 257, no. 5, pp. 877–881, May 2019, doi: 10.1007/S00417-019-04244-Y.

[13] L. Heskin, C. Simms, O. Traynor, and R. Galvin, “Designing a synthetic simulator to teach open surgical skills for limb exploration in trauma: a qualitative study exploring the experiences and perspectives of educators and surgical trainees,” *BMC Surg*, vol. 21, no. 1, Dec. 2021, doi: 10.1186/S12893-021-01417-7.

[14] B. S. Modibane, P. A. Owolawi, T. Mapayi, V. Malele, and G. Aiyetoro, “Virtual Reality Application Based on Inertia,” pp. 1–7, Dec. 2021, doi: 10.1145/3487923.3487931.

[15] N. Al-Hiyari and S. Jusoh, “The Current Trends of Virtual Reality Applications in Medical Education,” *Proceedings of the 12th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2020*, Jun. 2020, doi: 10.1109/ECAI50035.2020.9223158.

[16] M. Lorenz, C. Neupetsch, C. Rotsch, P. Klimant, and N. Hammer, “Early virtual reality user experience and usability assessment of a surgical shape memory alloy aspiration/irrigation instrument,” *26th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2019 - Proceedings*, pp. 1056–1057, Mar. 2019, doi: 10.1109/VR.2019.8798144.

[17] R. Sultanova and R. Sharaeva, “Virtual reality-based immersive simulation mechanics for invasive surgery training,” *Proceedings - International Conference on Developments in eSystems Engineering, DeSE*, vol. October-2019, pp. 924–928, Oct. 2019, doi: 10.1109/DESE.2019.00171.

[18] M. F. Obeid, R. Kelly, and F. D. McKenzie, “Development and Validation of a Hybrid Nuss Procedure Surgical Simulator and Trainer,” *IEEE Trans Biomed Eng*, vol. 68, no. 8, pp. 2520–2528, Aug. 2021, doi: 10.1109/TBME.2020.3048516.

[19] A. J. Lungu, W. Swinkels, L. Claesen, P. Tu, J. Egger, and X. Chen, “A review on the applications of virtual reality, augmented reality and mixed reality in surgical simulation: an extension to different kinds of surgery,” *Expert Rev*

Med Devices, vol. 18, no. 1, pp. 47–62, 2021, doi: 10.1080/17434440.2021.1860750.

[20] A. J. Lungu, W. Swinkels, L. Claesen, P. Tu, J. Egger, and X. Chen, "A review on the applications of virtual reality, augmented reality and mixed reality in surgical simulation: an extension to different kinds of surgery," *Expert Rev Med Devices*, vol. 18, no. 1, pp. 47–62, 2021, doi: 10.1080/17434440.2021.1860750.

[21] E. W. Riddle, D. Kewalramani, M. Narayan, and D. B. Jones, "Surgical Simulation: Virtual Reality to Artificial Intelligence," *Curr Probl Surg*, vol. 61, no. 11, p. 101625, Nov. 2024, doi: 10.1016/J.CPSURG.2024.101625.

[22] H. Wang, S. Jiang, and J. Wu, "A Virtual Reality Based Simulator for Training Surgical Skills in Procedure of Catheter Ablation," *International Conference on Artificial Intelligence and Virtual Reality*, pp. 247–248, Jul. 2018, doi: 10.1109/AIVR.2018.00057.

[23] E. Gardner, "Using Virtual Reality to Improve Perioperative Skills," *AORN J*, vol. 111, no. 3, pp. 281–283, Mar. 2020, doi: 10.1002/AORN.12964.

[24] B. Chahal, A. Aydin, and K. Ahmed, "Virtual reality vs. physical models in surgical skills training. An update of the evidence," *Curr Opin Urol*, vol. 34, no. 1, pp. 32–36, Jan. 2024, doi: 10.1097/MOU.0000000000001145.

[25] Y. Jung, V. Muddaluru, P. Gandhi, M. Pahuta, and D. Guha, "The Development And Applications Of Augmented And Virtual Reality Technology In Spine Surgery Training: A Systematic Review," *Can J Neurol Sci*, vol. 51, no. 2, pp. 255–264, Mar. 2024, doi: 10.1017/CJN.2023.46.

[26] Y. Jung, V. Muddaluru, P. Gandhi, M. Pahuta, and D. Guha, "The Development And Applications Of Augmented And Virtual Reality Technology In Spine Surgery Training: A Systematic Review," *Can J Neurol Sci*, vol. 51, no. 2, pp. 255–264, Mar. 2024, doi: 10.1017/CJN.2023.46.

[27] G. S. Goh, R. Lohre, J. Parvizi, and D. P. Goel, “Virtual and augmented reality for surgical training and simulation in knee arthroplasty,” *Arch Orthop Trauma Surg*, vol. 141, no. 12, pp. 2303–2312, Dec. 2021, doi: 10.1007/S00402-021-04037-1.

[28] E. Y. L. Teo *et al.*, “Immersive virtual reality as a surgical training tool: A cohort study,” *Heliyon*, vol. 10, no. 11, Jun. 2024, doi: 10.1016/J.HELIYON.2024.E31692.

[29] S. Syed Abdul, U. Upadhyay, D. Salcedo, and C. W. Lin, “Virtual reality enhancing medical education and practice: Brief communication,” *Digit Health*, vol. 8, Jan. 2022, doi: 10.1177/20552076221143948.

[30] F. Guerrini *et al.*, “The Use of Technology-Based Simulation among Medical Students as a Global Innovative Solution for Training,” *Brain Sci*, vol. 14, no. 7, p. 627, Jul. 2024, doi: 10.3390/BRAINSCI14070627/S1.

[31] E. Huang and X. (Phoenix) Chen, “Invited commentary: Virtual reality as a preoperative warm-up in surgical training,” *Am J Surg*, vol. 225, no. 6, pp. 953–954, Jun. 2023, doi: 10.1016/J.AMJSURG.2023.01.001.

[32] D. Selva-Ruiz and P. Martín-Ramallal, “Realidad virtual, publicidad y menores de edad:: otro reto de la cibersociedad ante las tecnologías inmersivas,” *Revista ICONO 14. Revista científica de Comunicación y Tecnologías emergentes*, vol. 17, no. 1, pp. 83–110, Jan. 2019, doi: 10.7195/ri14.v17i1.1234.

[33] U. De Holguín *et al.*, “Realidad Virtual: una tecnología al alcance de la universalización,” *Luz*, vol. 8, no. 2, pp. 1–12, 2009, Accessed: Mar. 23, 2025. [Online]. Available: https://www.redalyc.org/articulo.oa?id=589165875006

[34] S. C. Pérez, A. Muñoz, M. E. Stefanoni, D. B. Carbonari, and A. Muñoz, “Realidad virtual, aprendizaje inmersivo y realidad aumentada,” *XXIII Workshop de Investigadores en Ciencias de la Computación*, no. December, pp. 963–968, 2021, Accessed: Mar. 23, 2025. [Online]. Available: http://sedici.unlp.edu.ar/handle/10915/120930

[35] “(PDF) Sala de realidad virtual para la enseñanza médica Virtual Reality Room For Medical Teaching.” Accessed: Mar. 23, 2025. [Online]. Available: https://www.researchgate.net/publication/349948464_Sala_de_realidad_virtual_para_la_ensenanza_medica_Virtual_Reality_Room_For_Medical_Teaching

[36] “Realidad Aumentada y Realidad Virtual.” Accessed: Mar. 23, 2025. [Online]. Available: https://eduteka.icesi.edu.co/articulos/edutrends-realidad-virtual-y-aumentada

[37] J. Cabero Almenara, J. Barroso Osuna, and M. Obrador, “Realidad aumentada aplicada a la enseñanza de la medicina,” *Educación Médica*, vol. 18, no. 3, pp. 203–208, Jul. 2017, doi: 10.1016/J.EDUMED.2016.06.015.

[38] A. Canales Llantoy, L. Alfonso, and P. Vega, “Medieval Combat Con realidad virtual Item Type info:eu-repo/semantics/bachelorThesis.” [Online]. Available: http://hdl.handle.net/10757/621891

[39] “Desarrollo-y-evaluacion-de-un-IDE-en-realidad-virtual”.

[40] P. De Doctorado En Modelización, Y. Experimentación, E. N. Ciencia, and Y. Tecnología, “TESIS DOCTORAL Detección y clasificación automática de deficiencias en la visión del color mediante dispositivos de Realidad Virtual HALINA CARMEN CWIERZ LÓPEZ,” 2022.

[41] A. P. Pérez Negrón, J. L. D. Bonilla Carranza, L. F. Muñoz Mendoza, and M. A. Pérez Cisneros, *(PDF) Introducción al desarrollo de videojuegos con GODOT*, Primera edición. Jalisco, México: Astra Ediciones S. A. de C. V., 2022. Accessed: Apr. 10, 2025. [Online]. Available: https://www.researchgate.net/publication/368779614_Introduccion_al_desarrollo_de_videojuegos_con_GODOT

[42] C. González, A. Romo, J. Barrientos, and F. González, “LENGUAJES DE PROGRAMACION.”

[43] “PYPL PopularitY of Programming Language index.” Accessed: Apr. 10, 2025. [Online]. Available: https://pypl.github.io/PYPL.html

[44] N. I. Patricia Layedra Larrea, S. Alejandro Salazar Cazco III, M. I. Vinicio Ramos Valencia, and B. I. Alexander Baldeón Hermida, “Ciencias de la Educación Artículo de Investigación Escuela Superior Politécnica de Chimborazo, Ecuador,” vol. 8, no. 3, pp. 1601–1625, 2022, doi: 10.23857/dc.v8i3.

[45] “About — blender.org.” Accessed: Apr. 10, 2025. [Online]. Available: https://www.blender.org/about/

[46] “Comprar el software Autodesk Maya 2026 | Software de animación 3D.” Accessed: Apr. 10, 2025. [Online]. Available: https://www.autodesk.com/mx/products/maya/overview

[47] “Cinema 4D | Software de animación y modelado 3D - Maxon.” Accessed: Apr. 10, 2025. [Online]. Available: https://www.maxon.net/es/cinema-4d?srsltid=AfmBOorJjPDXDOGOkqwVVQgi3N-L_xIcSJtLrjpphYdXSE_TEtLWC0PB

[48] J. Jerald, *The VR Book: Human-Centered Design for Virtual Reality*. Association for Computing Machinery and Morgan \& Claypool, 2015. doi: 10.1145/2792790.

[49] M. Slater and M. V. Sanchez-Vives, “Enhancing our lives with immersive virtual reality,” Dec. 01, 2016, *Frontiers Media S.A.* doi: 10.3389/frobt.2016.00074.

[50] “Unity - Manual: System requirements for Unity 6.1.” Accessed: May 11, 2025. [Online]. Available: https://docs.unity3d.com/Manual/system-requirements.html

[51] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt, “A systematic review of immersive virtual reality applications for higher education:

Design elements, lessons learned, and research agenda," *Comput Educ*, vol. 147, Apr. 2020, doi: 10.1016/j.compedu.2019.103778.

[52] G. Makransky and L. Lilleholt, "A structural equation modeling investigation of the emotional value of immersive virtual reality in education," *Educational Technology Research and Development*, vol. 66, no. 5, pp. 1141–1164, Oct. 2018, doi: 10.1007/S11423-018-9581-2.

[53] W. W. Royce, "Managing the Development of Large Software Systems (1970)," *Ideas That Created the Future*, pp. 321–332, Feb. 2021, doi: 10.7551/MITPRESS/12274.003.0035.

[54] P. Kruchten, "THE RATIONAL UNIFIED PROCESS AN INTRODUCTION THIRD EDITION," pp. 1–309, 2004.

[55] B. Kevin Forsberg, "The Relationship of System Engineering to the Project Cycle," pp. 9–11, 1994.

[56] B. W. Boehm, "A Spiral Model of Software Development and Enhancement."

[57] "Manifesto for Agile Software Development." Accessed: Apr. 29, 2025. [Online]. Available: https://agilemanifesto.org/

[58] "Home | Scrum Guides." Accessed: Apr. 29, 2025. [Online]. Available: https://scrumguides.org/

[59] K. Beck and C. Andres, "Extreme Programming Explained: Embrace Change, 2nd Edition (The XP Series)," vol. 2, p. 46, 2004.

[60] J. Boeg, "Priming Kanban," *InfoQ Enterprise Software Development Series*, p. 43, 2012, Accessed: Apr. 29, 2025. [Online]. Available: https://www.infoq.com/minibooks/priming-kanban-jesper-boeg

[61] Mary. Poppendieck and Tom. Poppendieck, "Lean software development : an agile toolkit," p. 203, 2003.

[62] J. V. Tuapanta Dacto, M. Duque Vaca, and ÁngelMena Reinoso, “ALFA DE CRONBACH PARA VALIDAR UN CUESTIONARIO DE USO DE TIC EN DOCENTES UNIVERSITARIOS ,” *Revista mktDescubre - ESPOCH FADE*, pp. 37–48, Dec. 2017, Accessed: Apr. 02, 2025. [Online]. Available: https://core.ac.uk/download/pdf/234578641.pdf

[63] S. Monterroza Linconl, “Atlas.Ti: Herramienta de Análisis para Potenciar el Desarrollo del Pensamiento Espacial con GeoGebra,” *Ciencia Latina Revista Científica Multidisciplinar*, vol. 8, no. 2, pp. 6102–6123, May 2024, doi: 10.37811/cl_rcm.v8i2.11025.

[64] P. Sirimanna and M. A. Gladman, “Development of a proficiency-based virtual reality simulation training curriculum for laparoscopic appendicectomy,” *ANZ J Surg*, vol. 87, no. 10, pp. 760–766, Oct. 2017, doi: 10.1111/ANS.14135.

[65] H. J. Kim *et al.*, “Immersive virtual reality simulation training for cesarean section: a randomized controlled trial,” 2023, doi: 10.1097/JS9.0000000000000843.

[66] N. Gong and M. Gong, “MiRNA-221 from tissue may predict the prognosis of patients with osteosarcoma,” *Medicine (United States)*, vol. 97, no. 29, Jul. 2018, doi: 10.1097/MD.0000000000011100,.

[67] H. J. Kim *et al.*, “Immersive virtual reality simulation training for cesarean section: a randomized controlled trial,” *Int J Surg*, vol. 110, no. 1, p. 194, Jan. 2023, doi: 10.1097/JS9.0000000000000843.

[68] J. Zabaleta *et al.*, “Clinical trial on nurse training through virtual reality simulation of an operating room: assessing satisfaction and outcomes,” *Cirugía Española (English Edition)*, vol. 102, no. 9, pp. 469–476, Sep. 2024, doi: 10.1016/j.cireng.2024.04.012.

[69] H. Y. Kim, “Development and Usability Assessment of Virtual Reality- and Haptic Technology-Based Educational Content for Perioperative Nursing

Education," *Healthcare 2024, Vol. 12, Page 1947*, vol. 12, no. 19, p. 1947, Sep. 2024, doi: 10.3390/HEALTHCARE12191947.

[70] T. Li *et al.*, "Using Virtual Reality to Enhance Surgical Skills and Engagement in Orthopedic Education: Systematic Review and Meta-Analysis.," *J Med Internet Res*, vol. 27, no. 1, p. e70266, May 2025, doi: 10.2196/70266.

[71] "Unity 3D for eLearning Game Development : Articles | The Learning Guild." Accessed: Jun. 25, 2025. [Online]. Available: https://www.learningguild.com/articles/unity-3d-for-elearning-game-development/

[72] V. Agrahari and S. Chimalakonda, "What's Inside Unreal Engine? - A Curious Gaze!," *ACM International Conference Proceeding Series*, Feb. 2021, doi: 10.1145/3452383.3452404.

[73] E. A. Keshner, P. T. Weiss, D. Geifman, and D. Raban, "Tracking the evolution of virtual reality applications to rehabilitation as a field of study," *J Neuroeng Rehabil*, vol. 16, no. 1, pp. 1–15, Jun. 2019, doi: 10.1186/S12984-019-0552-6/TABLES/4.

[74] "CS 148 - Introduction To Computer Graphics and Imaging | PDF | Rendering (Computer Graphics) | Texture Mapping." Accessed: Jun. 25, 2025. [Online]. Available: https://es.scribd.com/document/640395797/CS-148-Introduction-to-Computer-Graphics-and-Imaging

[75] I. Yenisel Molina Hernández, D. Ailec Granda Dihigo, and A. Velázquez Cintra, "The non-functional software requirements. A strategy for its development in the Medical Informatics Center," *Revista Cubana de Ciencias Informáticas*, vol. 13, no. 2, 2019, Accessed: Apr. 08, 2025. [Online]. Available: http://rcci.uci.cu

[76] I. Sommerville, *INGENIERÍA DE SOFTWARE*, NOVENA EDICIÓN, 2011. México: PEARSON EDUCACIÓN, 2011. Accessed: Apr. 08, 2025. [Online]. Available:

https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf

[77] “Systems and software engineering-Life cycle processes-Requirements engineering,” 2018. [Online]. Available: www.iso.org

[78] “Cyclomatic complexity: Definition and limits in understanding code quality.” Accessed: May 12, 2025. [Online]. Available: https://getdx.com/blog/cyclomatic-complexity/

[79] “Understanding Cyclomatic Complexity: A Developer’s Comprehensive Guide | by typo | The Typo Diaries | Medium.” Accessed: May 12, 2025. [Online]. Available: https://medium.com/beyond-the-code-by-typo/understanding-cyclomatic-complexity-a-developers-comprehensive-guide-820772732514

[80] “How to Use SonarQube to Improve Your Code Quality.” Accessed: May 12, 2025. [Online]. Available: https://www.freecodecamp.org/news/how-to-improve-your-code-quality-with-sonarqube/

[81] G. Gazc´, Gazcón N, and Soto M, “Visual Studio Analyzer Computací on GráficaGr´Gráfica, 30 de Abril de 2014”.

[82] “Halstead’s Software Metrics – Software Engineering | GeeksforGeeks.” Accessed: May 12, 2025. [Online]. Available: https://www.geeksforgeeks.org/software-engineering-halsteads-software-metrics/

[83] “What Metrics Does Understand Have? : SciTools Support.” Accessed: May 12, 2025. [Online]. Available: https://support.scitools.com/support/solutions/articles/70000582223-what-metrics-does-understand-have-

[84] “Cppcheck - A tool for static C/C++ code analysis.” Accessed: May 12, 2025. [Online]. Available: https://cppcheck.sourceforge.io/

[85] C. Anthes, R. J. García-Hernández, M. Wiedemann, and D. Kranzlmüller, “State of the art of virtual reality technology,” *IEEE Aerospace Conference Proceedings*, vol. 2016-June, Jun. 2016, doi: 10.1109/AERO.2016.7500674.

[86] “Frames Per Second (FPS) Definition | Encord.” Accessed: May 17, 2025. [Online]. Available: https://encord.com/glossary/frames-per-second-fps-definition/

[87] “Unity - Manual: Unity Profiler.” Accessed: May 17, 2025. [Online]. Available: https://docs.unity3d.com/Manual/Profiler.html

[88] “Optimiza el rendimiento y la calidad | Herramientas para la generación de perfiles de Unity.” Accessed: May 17, 2025. [Online]. Available: https://unity.com/es/features/profiling

[89] “Monitor Performance with OVR Metrics Tool | Meta Horizon OS Developers.” Accessed: May 17, 2025. [Online]. Available: https://developers.meta.com/horizon/documentation/native/android/ts-ovrmetricstool/

[90] “Unreal Insights Overview | Unreal Engine 4.27 Documentation | Epic Developer Community.” Accessed: May 17, 2025. [Online]. Available: https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-insights-overview?application_version=4.27

[91] “Unreal Insights in Unreal Engine | Unreal Engine 5.5 Documentation | Epic Developer Community.” Accessed: May 17, 2025. [Online]. Available: https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-insights-in-unreal-engine

[92] S. Mittal and J. S. Vetter, “A survey of CPU-GPU heterogeneous computing techniques,” *ACM Comput Surv*, vol. 47, no. 4, Jul. 2015, doi: 10.1145/2788396.

[93] “RenderDoc.” Accessed: May 16, 2025. [Online]. Available: https://renderdoc.org/

[94] “NVIDIA Nsight Graphics | NVIDIA Developer.” Accessed: May 16, 2025. [Online]. Available: https://developer.nvidia.com/nsight-graphics

[95] T. Tullis and B. Albert, “Combined and Comparative Metrics,” *Measuring the User Experience*, pp. 187–208, 2013, doi: 10.1016/B978-0-12-415781-1.00008-X.

[96] “Método NASA-TLX.”

[97] J. Bustamante, “La herramienta de tests de usabilidad a distancia Loop11,” *Profesional de la información, ISSN-e 1699-2407, ISSN 1386-6710, Vol. 19, Nº 4, 2010 (Ejemplar dedicado a: Arquitectura de la información), págs. 425-430*, vol. 19, no. 4, pp. 425–430, 2010, Accessed: May 24, 2025. [Online]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=3274652&info=resumen&idioma=ENG

[98] “Historias de usuario | Ejemplos y plantilla | Atlassian.” Accessed: Jun. 11, 2025. [Online]. Available: https://www.atlassian.com/es/agile/project-management/user-stories

[99] G. Booch, J. Rumbaugh, and I. Jacobson, “The Unified Modeling Language User Guide Second Edition,” *Pearson Education, Inc.*, p. 496, 2005.

[100] M. Fowler, “UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition: Proquest Tech & Business Books,” p. 208, 2003, Accessed: Apr. 29, 2025. [Online]. Available: http://proquestcombo.safaribooksonline.com.lcproxy.shu.ac.uk/book/software-engineering-and-development/uml/0321193687

[101] Craig. Larman, “Applying UML and patterns : an introduction to object-oriented analysis and design and iterative development,” p. 703, 20092005.

[102] Y. Ding and others, “User-centered requirements engineering for surgical training in virtual reality environments,” *J Biomed Inform*, vol. XXX, no. X, pp. XXX–XXX, 2022.

[103] Y. Sun and others, “Designing immersive virtual environments for nursing education: A UX approach,” *Nurse Educ Today*, vol. XXX, no. X, pp. XXX–XXX, 2021.

[104] M. Haque and others, “A VR-based training system for minimally invasive surgery: software architecture and implementation challenges,” *Int J Med Inform*, vol. XXX, no. X, pp. XXX–XXX, 2020.

[105] C. McGrath and others, “Evaluating the fidelity and usability of a virtual reality simulation for emergency nursing,” *Clin Simul Nurs*, vol. XXX, no. X, pp. XXX–XXX, 2019.

[106] A. Li and others, “From prototype to practice: Implementing VR-based tools in hospital-based training,” *Simulation in Healthcare*, vol. XXX, no. X, pp. XXX–XXX, 2021.

[107] D. Aspinall and others, “Maintaining long-term clinical training platforms based on VR: updates, bugs, and scalability,” *J Med Syst*, vol. XXX, no. X, pp. XXX–XXX, 2018.

[108] A. Cataldo, “Design Science Research (DSR): Una breve introducción.”

Anexo 1: Encuesta

1 ¿A usted le gusta usar la tecnología?

- Si
- No
- Un poco

2 ¿Qué experiencia tiene con el uso de sistemas de realidad virtual para fines educativos o laborales?

- Experto/a
- Alguna experiencia
- Ninguna experiencia

3 ¿Qué tan útil considera que sería la realidad virtual para mejorar la identificación de instrumentos quirúrgicos en su formación?

- Muy útil
- Útil
- Poco útil
- Nada útil

4 ¿Cree que las simulaciones en realidad virtual podrían complementar la practica en un laboratorio físico?

- Totalmente de acuerdo
- De acuerdo
- En desacuerdo

- Totalmente en desacuerdo

5 ¿Qué tan fácil cree que le resultaría aprender el correcto orden del armado de la mesa Mayo en un entorno de realidad virtual?

- Muy fácil
- Fácil
- Difícil
- Muy difícil

6 ¿Cree que el uso de la realidad virtual podría mejorar su confianza al preparar e identificar los instrumentos de una mesa Mayo en el mundo real?

- Totalmente de acuerdo
- De acuerdo
- En desacuerdo
- Totalmente en desacuerdo

7 ¿Qué tan necesario considera que es desarrollar un sistema de realidad virtual específico para practicas quirúrgicas en la FCS?

- Muy necesario
- Necesario
- Nada necesario

8 ¿Considera que los sistemas de realidad virtual podría ser una solución viable para suplir la capacidad limitada de un laboratorio físico?

- Totalmente de acuerdo

- De acuerdo
- En desacuerdo
- Totalmente en desacuerdo

9 ¿Considera factible que los estudiantes se adapten al uso de sistemas de realidad virtual para su formación académica?

- Totalmente de acuerdo
- De acuerdo
- En desacuerdo
- Totalmente en desacuerdo

10 ¿Cree que un sistema de realidad virtual podría facilitar la práctica autónoma fuera del laboratorio?

- Totalmente de acuerdo
- De acuerdo
- En desacuerdo
- Totalmente en desacuerdo

11 ¿Que tan importante considera que sea la implementación tecnología de realidad virtual para solventar la falta de acceso a prácticas en laboratorios?

- Muy útil
- Útil
- Poco útil
- Nada útil

12 ¿Cree que un sistema de realidad virtual permitiría una mejor distribución del tiempo de practica en comparación con los laboratorios tradicionales?

- Totalmente de acuerdo
- De acuerdo
- En desacuerdo
- Totalmente en desacuerdo

13 ¿Estaría dispuesto a utilizar un sistema de realidad virtual si estuviera disponible en su institución?

- Si
- No
- Un poco

14 ¿Cree que la implementación de un sistema de realidad virtual mejorara significativamente la formación profesional en la FCS?

- Totalmente de acuerdo
- De acuerdo
- En desacuerdo
- Totalmente en desacuerdo

15 ¿Considera que el desarrollo de este sistema debería involucrar opiniones de estudiantes, profesores y expertos en cirugía?

- Si
- No
- No estoy seguro

16 ¿Al terminar la encuesta como se sintió?

- Feliz
- Aburrido
- Triste
- Confundido

Anexo 2: Entrevistas

Entrevista a realizar por parte del experto en desarrollo

1 ¿Cuáles son las principales consideraciones técnicas al desarrollar un simulador de realidad virtual para el entrenamiento médico?

2 ¿Qué motor gráfico y herramientas recomendaría para crear simulaciones médicas altamente realistas?

3 ¿Qué criterios utiliza para evaluar la fidelidad visual y funcionalidad de una simulación en RV?

4 ¿Qué metodologías de desarrollo ágil o tradicional son más efectivas para proyectos como este?

5 ¿Qué aspectos de la interacción con pacientes reales considera que son difíciles de replicar en una simulación de RV?

Entrevista a realizar por parte del experto e cirugía

1 ¿Ha tenido experiencia utilizando simuladores de realidad virtual en la enseñanza, de ser así, ¿nos podría explicar cuáles fueron sus resultados obtenidos?

2 ¿Usted piensa que es necesario implementar un sistema de realidad virtual en la universidad para mejorar el aprendizaje en los estudiantes?

3 ¿Cómo considera que la realidad virtual podría impactar en la confianza y en las habilidades obtenidas al pasar del tiempo en los estudiantes?

4 ¿Qué beneficios cree que la realidad virtual aporte en el aprendizaje de los estudiantes?

5 ¿Usted considera importante incluir aspectos como el estrés o la presión del tiempo en las simulaciones?

6 ¿Qué tan efectivo usted considera que la simulación de realidad virtual mejora o en comparación de los métodos tradicionales que maneja?

7 ¿Usted cree que los estudiantes podrían depender demasiado de la tecnología y no desarrollar adecuadamente sus habilidades?

8 ¿Qué características usted consideraría esenciales para una simulación de realidad virtual?

9 ¿Consideran que los escenarios, las simulaciones, los personajes, tiene que ser exactamente al mundo real para que el estudiante pueda tener una mejor eficiencia en el aprendizaje?

Anexo 3: Encuesta de Usabilidad

CUESTIONARIO DE USABILIDAD - ESCALA SUS

Instrucciones: A continuación, encontrará una serie de afirmaciones sobre el sistema que acaba de utilizar. Indique su nivel de acuerdo con cada afirmación marcando una sola opción en la escala de 1 a 5, donde:

1 = Totalmente en desacuerdo

2 = En desacuerdo

3 = Neutral

4 = De acuerdo

5 = Totalmente de acuerdo

1. 1. Me gustaría usar este sistema con frecuencia.

   [1] [2] [3] [4] [5]

2. 2. Encontré el sistema innecesariamente complejo.

   [1] [2] [3] [4] [5]

3. 3. Consideré que el sistema era fácil de usar.

   [1] [2] [3] [4] [5]

4. 4. Creo que necesitaría la ayuda de un experto para usar este sistema.

   [1] [2] [3] [4] [5]

5. 5. Las funciones del sistema estaban bien integradas.

   [1] [2] [3] [4] [5]

6. 6. El sistema era inconsistente.

[1] [2] [3] [4] [5]

7. 7. Creo que la mayoría de las personas aprenderían a usar este sistema muy rápidamente.

[1] [2] [3] [4] [5]

8. 8. Consideré que el sistema fue difícil de usar.

[1] [2] [3] [4] [5]

9. 9. Me sentí seguro usando este sistema.

[1] [2] [3] [4] [5]

10. 10. Necesité aprender muchas cosas antes de poder usar el sistema.

[1] [2] [3] [4] [5]

Observaciones adicionales del usuario:

_____

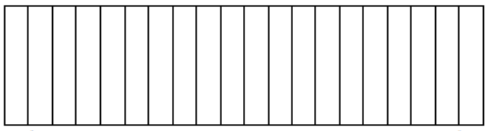
_____

_____

Anexo 4: Cuestionario del método NASA TLX (Task Load Index)

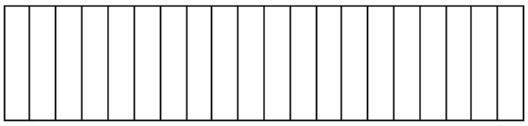
◯ Exigencias Mentales ◯ Exigencias Físicas	◯ Exigencias Físicas ◯ Exigencias Temporales	◯ Exigencias Temporales ◯ Esfuerzo
◯ Exigencias Mentales ◯ Exigencias Temporales	◯ Exigencias Físicas ◯ Rendimiento	◯ Exigencias Temporales ◯ Nivel de Frustración
◯ Exigencias Mentales ◯ Rendimiento	◯ Exigencias Físicas ◯ Esfuerzo	◯ Rendimiento ◯ Esfuerzo
◯ Exigencias Mentales ◯ Esfuerzo	◯ Exigencias Físicas ◯ Nivel de Frustración	◯ Rendimiento ◯ Nivel de Frustración
◯ Exigencias Mentales ◯ Nivel de Frustración	◯ Exigencias Temporal ◯ Rendimiento	◯ Esfuerzo ◯ Nivel de Frustración

Exigencia Mental. ¿Qué tan demandante mentalmente es la tarea?



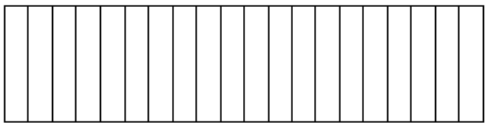
Baja Alta

Exigencia Física. ¿Qué tan demandante mentalmente es la tarea?



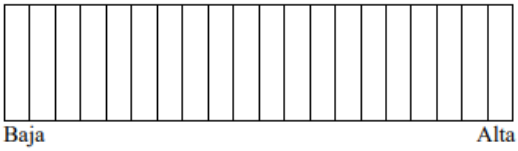
Baja Alta

Exigencias Temporales. ¿Qué tan fuerte o rápido es el ritmo impuesto para hacer la tarea?



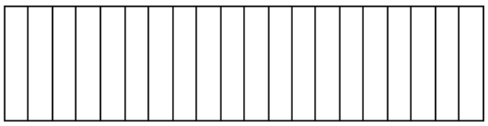
Baja Alta

Rendimiento. ¿Qué tan exitoso ha sido para lograr lo que ha querido?



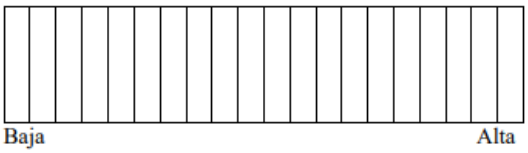
Baja Alta

Esfuerzo. ¿Qué tan duro tiene que trabajar para lograr un adecuado nivel de rendimiento?



Baja Alta

Nivel de Frustración. ¿Qué tan inseguro, irritado o estresado y molesto está por la tarea?



Baja Alta

Anexo 5: Metodo de evaluación TAM

Instrucciones: A continuación, se presentan afirmaciones sobre la metodología de desarrollo de RV. Indique su nivel de acuerdo usando la siguiente escala:

Escala Likert	Significado
1	Totalmente en desacuerdo
2	En desacuerdo
3	Neutral
4	De acuerdo
5	Totalmente de acuerdo

Sección 1: Utilidad Percibida (UP)

Evalúa si la metodología contribuye al logro de objetivos clínicos y educativos.

Ítem	Afirmación
UP1	La metodología mejora la calidad de los escenarios clínicos simulados.
UP2	La metodología permite diseñar simulaciones quirúrgicas más realistas.
UP3	El uso de esta metodología facilita el aprendizaje práctico de los estudiantes.
UP4	La metodología mejora la eficiencia en el desarrollo de aplicaciones de RV.
UP5	Implementar esta metodología aporta beneficios clínico-educativos evidentes.

Sección 2: Facilidad de Uso Percibida (FUP)

Evalúa qué tan sencilla y comprensible es la metodología para los involucrados.

Ítem	Afirmación
FUP1	Es fácil comprender las fases y procesos de la metodología.
FUP2	Los roles de cada actor (médico, desarrollador, estudiante) están claramente definidos.
FUP3	La metodología puede ser aplicada sin conocimientos técnicos avanzados en ingeniería.
FUP4	La documentación proporcionada es suficiente y clara.
FUP5	La implementación no requiere grandes recursos adicionales.

Sección 3: Actitud hacia el uso (ATU)

Evalúa la disposición emocional y racional del usuario hacia la metodología.

Ítem	**Afirmación**
ATU1	Me resulta agradable trabajar con esta metodología.
ATU2	Considero positivo que esta metodología se utilice en educación clínica.
ATU3	Me gustaría que la metodología se adopte en otros proyectos similares.

Sección 4: Intención de Uso (IU)

Evalúa la probabilidad de adopción futura.

Ítem	**Afirmación**
IU1	Recomendaría el uso de esta metodología a otros profesionales.
IU2	Estoy dispuesto a seguir usando esta metodología en futuros proyectos.
IU3	Considero esta metodología como estándar viable en el desarrollo de simulaciones clínicas.

Anexo 6: Manual de instalación del sistema

SISTEMA NURSETOOLS XR

DESCRIPCIÓN DEL SISTEMA

NurseTools XR es una simulación de un quirófano en realidad virtual que permite:

- Visualizar el nombre y ubicación del mobiliario quirúrgico.
- Realizar el armado del instrumental quirúrgico en la mesa Mayo.
- Simular dos tipos de operaciones: cesárea y apendicectomía, con sus respectivos tiempos quirúrgicos.

REQUISITOS MÍNIMOS DEL SISTEMA

Recurso	**Requisitos mínimos**	**Recomendado**
Procesador	Intel Core i5 / Ryzen 5	Intel Core i7 / Ryzen 7
Tarjeta gráfica	NVIDIA GTX 1050 Ti / AMD RX 580	Modelos superiores
Memoria RAM	8 GB	16 GB
Sistema operativo	Windows o MacOS	Windows
Gafas de realidad virtual	Meta Quest 3 / 3s	Meta Quest 3 / 3s
Cables de conexión	USB – Tipo C o Tipo c – Tipo C	USB – Tipo C 0 Tipo c – Tipo C

INSTALACIÓN DE META QUEST DEVELOPER HUB

Paso 1

- Descarga Meta Quest Developer Hub según tu sistema operativo: Windows MacOS

Paso 2

- Ejecuta el instalador y sigue las instrucciones hasta completar la instalación.

Paso 3

- Abre el programa y dirígete a la pestaña Device Manager.

Paso 4

- Encender tus gafas Meta Quest.

Paso 5

- Conecta tus gafas Meta Quest a la computadora usando un cable USB – Tipo C o Tipo c – Tipo C.
- Nota: Si tiene algún problema a instalar revisar el siguiente video: https://youtu.be/PWA9tYmGjrI

Si no aparecen en la pestaña Device Manager tus gafas Meta Quest, es necesario activar el modo desarrollador.

ACTIVAR EL MODO DESARROLLADOR EN META QUEST

Paso 1

- En tu celular, abre la Play Store o App Store y descarga la aplicación Meta Horizon.

Paso 2

- Crea una cuenta o inicia sesión. Luego, vincula tus gafas a la app siguiendo los pasos del siguiente video: https://youtu.be/hEz0_qlCEP0?list=LL&t=329

Paso 3

- Una vez vinculadas tus gafas Meta Quest activamos el modo desarrollador siguiendo los pasos del siguiente video: https://youtu.be/guN46exIaeI?list=LL

Paso 4

- Cierra y vuelve a abrir Meta Quest Developer Hub en tu PC.
- Vuelve a conectar las gafas; ahora deberían aparecer en el Device Manager.

INSTALACIÓN DEL SISTEMA NURSETOOLS XR

Paso 1

- Descarga el archivo APK llamado NurseTools XR: https://drive.google.com/drive/folders/1d0MYnn6bUpdPX8xo61z3bTJKMIRBX7pk?usp=sharing

Paso 2

- Abre Meta Quest Developer Hub y accede a Device Manager.

Paso 3

- Conecta tus gafas a la computadora y verifica que sean reconocidas.

Paso 4

- Dirígete a la pestaña Apps y arrastra allí el archivo APK.

Paso 5

- Selecciona la opción Connected Device: [Nombre de tus gafas.

Paso 6

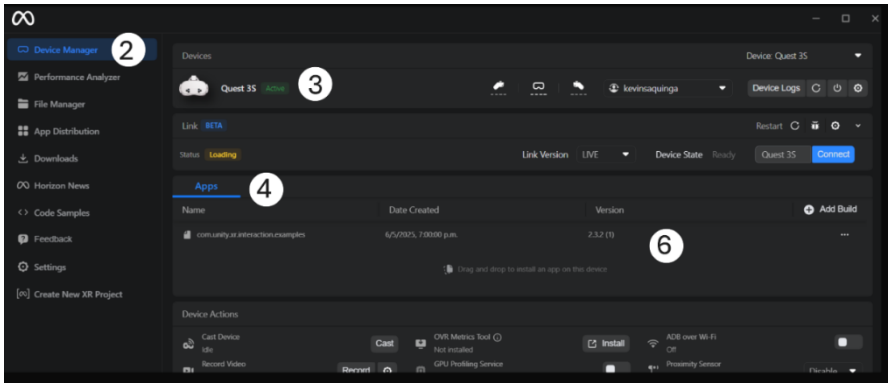
- El sistema se instalará automáticamente. Podrás verificarlo en la sección de **Apps** del dispositivo.

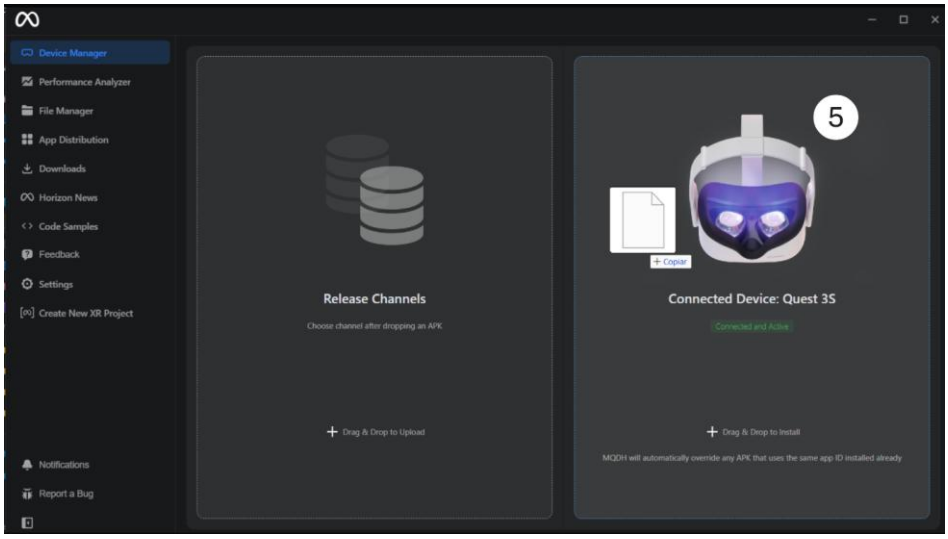
Paso 7

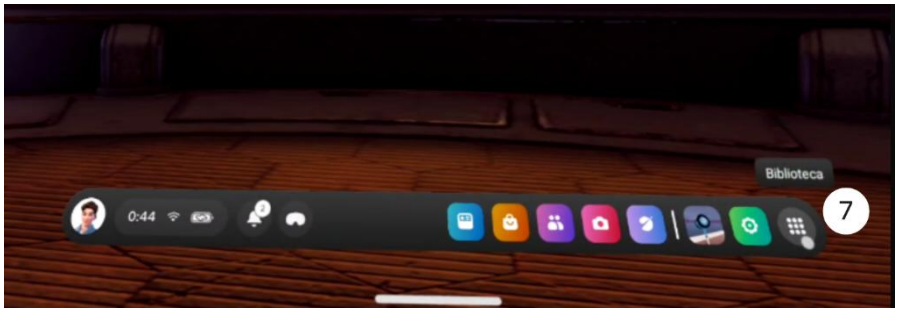
- Colócate las gafas Meta Quest, entra a **Biblioteca > Orígenes desconocidos** y ejecuta el sistema NurseTools XR.

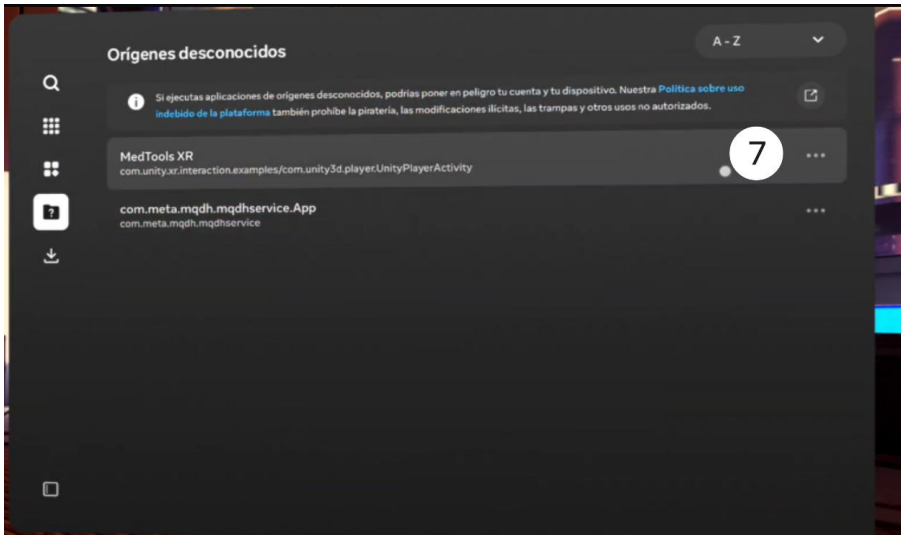
Paso 8

Una vez abierto por primera vez, el sistema quedará disponible directamente en el menú principal de las gafas.









FIN DEL MANUAL DE INSTALACIÓN

Anexo 7: Manual de Usuario

APLICACIÓN DE REALIDAD VIRTUAL PARA PRÁCTICAS DE ENFERMERÍA

NURSETOOLS

Introducción

Este manual de usuario está diseñado para guiar a los estudiantes y docentes en el uso correcto de la aplicación, facilitando la comprensión de su funcionamiento y el aprovechamiento de sus funcionalidades. La aplicación incluye lo que es la interacción con un cuarto de operación en el cual se pueden reconocer y ubicar el equipamiento de este y además una simulación de cómo debe de ordenar las diferentes herramientas en la mesa de mayo.

Requisitos del sistema

- Meta Quest 3S/3
- Computador: con Intel i3 o ryzen 3
- Memoria: Ram 8 GB+
- Sistema operative: Windows 10 o mayor, MacOS Sierra 10 o mayor
- Cable USB-C

Controles y navegación en VR

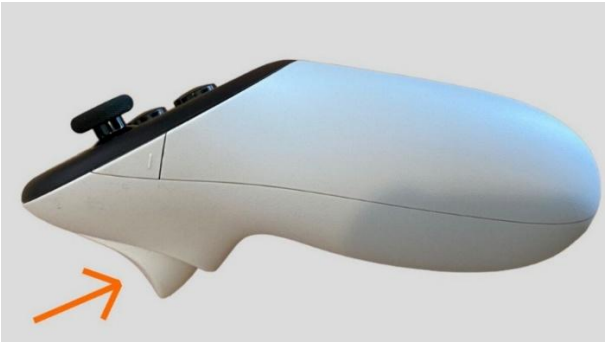
- **Movimiento del usuario**



Joystick izquierdo: Permite moverse libremente dentro del entorno virtual (adelante, atrás, izquierda, derecha).

Joystick derecho: Permite girar la vista hacia la izquierda o derecha, facilitando la orientación sin necesidad de girar físicamente el cuerpo.

Interacción



Botón Trigger (gatillo) izquierdo o derecho: Sirve para seleccionar objetos o interactuar con elementos del entorno. Al apuntar a un objeto con el rayo de selección activado, presionar el trigger lo activará o abrirá según el contexto.

Rayo de selección



Botón X (controlador izquierdo) y Botón A (controlador derecho): Al presionarlos, se activa un rayo láser de selección desde la mano correspondiente. Este rayo permite apuntar con precisión a objetos o elementos interactivos a distancia.

Consejos de navegación

El desplazamiento mediante joystick es suave, pero se recomienda realizar pausas si se experimenta mareo. Siempre revise su entorno físico para evitar colisiones con objetos reales al moverse. Interfaz de usuario Menú Principal

Interfaz de usuario Menú Principal

Elementos del menú principal

1. **Botón de Inicio (>):** Avanzar a la Escena Quirófano.
2. **Botón de Tutorial:** Iniciar Tutorial de controles.

3. **Botón de Salida (X):** Cerrar de la aplicación.



Escenario Quirófano Simulado

Interacción con el escenario

Si se apunta con el rayo de interacción a un objeto se mostrará el nombre del mismo

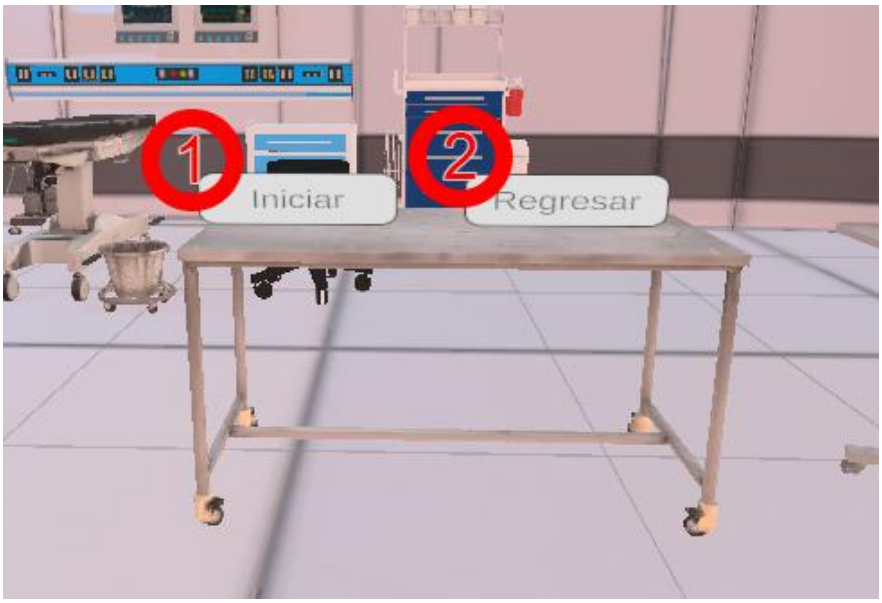


Simulación de armado de la mesa mayo

Elementos del escenario quirófano

1. **Botón Iniciar:** Inicia la simulación mostrando el menú de selección de operación y tiempo.

2. **Botón Regresar:** Regresa al menú principal



3. **Menú desplegable tipo de operación:** Selección del tipo de operación.

4. **Menú desplegable tiempos de Operación:** Selección del tiempo de la operación.

5. **Botón Cerrar(X):** Cierra el menú de Operación.

6. **Botón General:** Inicia la simulación general de todos los tiempos de la operación seleccionada.

7. **Botón de Inicio(>):** Inicia la práctica según el tipo de operación y el tiempo seleccionados

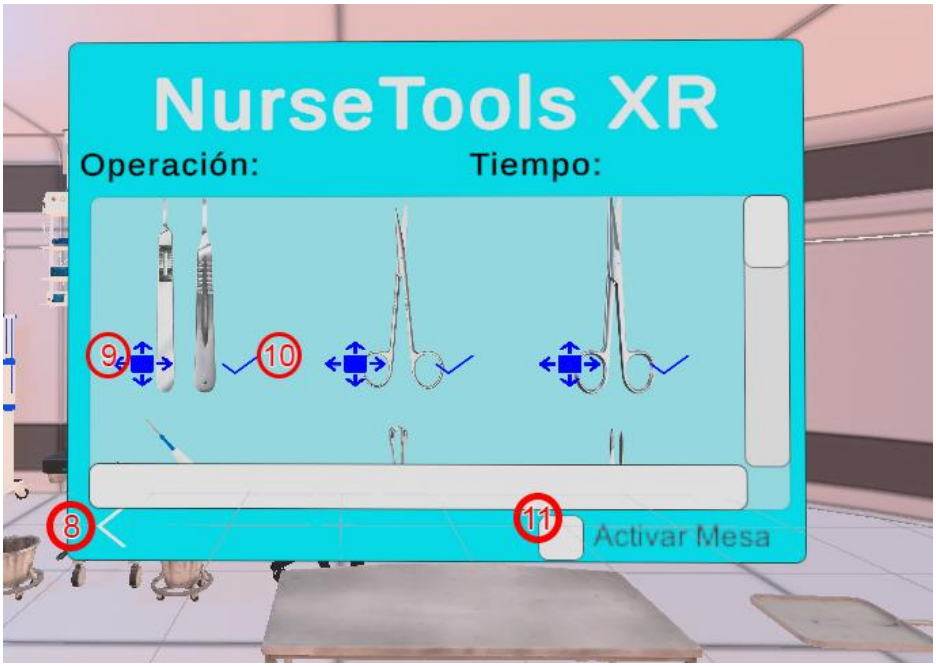


8. **Boton Regresar(<):** Vuelve al menu anterior

9. **Boton Expandir:** Expande la imagen del instrumento en otro ventana en el escenario.

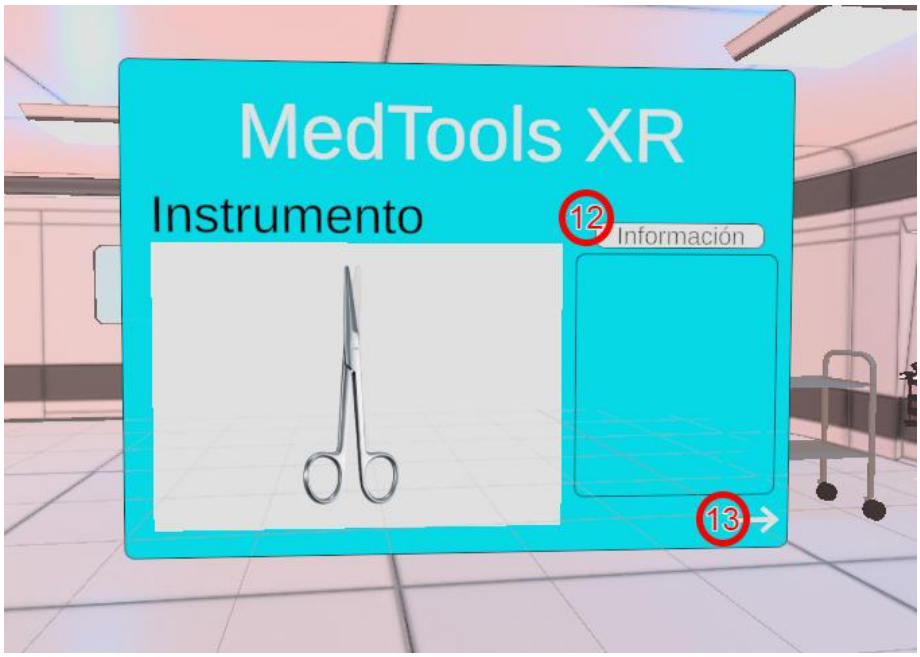
10. **Boton Seleccionar:** Selecciona el instrumento y si el instrumento es correcto según el tiempo de la operación se lo mostrara en la mesa de mayo simulada.

11. **Check Activar Mesa de mayo:** Muestra la mesa de mayo simulada en el escenario.

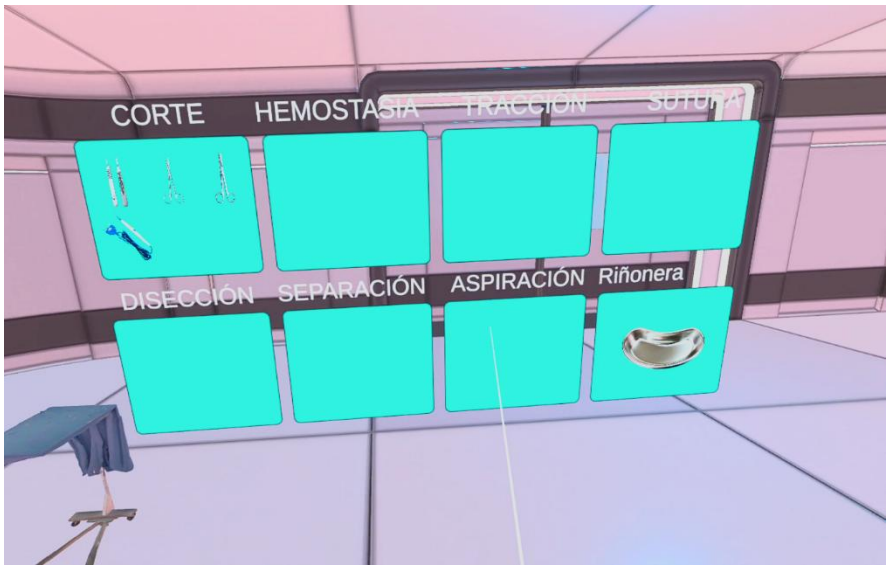


12. **Botón información:** Muestra el nombre y la información de instrumento.

13. **Botón cerrar (->):** Cierra la ventana información.



Mesa de mayo simulada



Retroalimentación y evaluación

En la simulación de entrenamiento por tiempo específico cuando se seleccione un instrumento incorrecto según su tipo y tiempo de operación se mostrará un mensaje de advertencia.



En la simulación general cuando se termine la misma se mostrará el total de errores del estudiante y el tiempo total transcurrido desde el que empezó la simulación.

