



Análisis de la calidad de proyecto de software RVivum bajo el marco de la norma ISO 25010:2023

Vicente Gaete Olmos

vicente.gaeteo@usm.cl

Marcello Visconti

Profesor Guía

Resumen: Este trabajo aborda la problemática de la pérdida de acceso al patrimonio cultural destruido, deteriorado o inaccesible en Chile, lo que limita su preservación, difusión y uso educativo. En este contexto, se presenta el proyecto de software RVivum, una aplicación de realidad virtual que permite explorar espacios patrimoniales digitales, reconstruidos mediante inteligencia artificial generativa. Para que el software cumpla con el propósito este debe cumplir con estándares de calidad de software. El objetivo de este trabajo es proponer un plan de mejora de la calidad del software, a partir de una evaluación estructurada de su arquitectura y de sus atributos de calidad. Para esto, se identifican y priorizan los atributos de calidad más relevantes, se aplica la metodología ATAM para analizar las decisiones arquitectónicas e identificar sus tradeoffs, y se evalúa la calidad del sistema utilizando el modelo de calidad de producto basado en la norma 25010:2023 mediante métricas definidas. Se identifican las características de Adecuación funcional, Eficiencia de desempeño, Capacidad de interacción, Fiabilidad y Mantenibilidad. Los resultados muestran falencias principalmente en la precisión de interacciones, la estabilidad de la tasa de FPS, el rendimiento de GPU y la tolerancia a fallos. A partir del análisis se proponen mejoras específicas enfocadas a cada característica. Se propone la incorporación de nuevos componentes C#, la estandarización del flujo de integración de recursos y la implementación de mecanismos de optimización como Occlusion Culling, FFR y carga asíncrona de escenas. El impacto esperado de este trabajo es fortalecer la viabilidad técnica de RVivum y de aplicaciones de realidad virtual similares.

Palabras Clave: Realidad virtual, Calidad de Software, Atributos de Calidad, ATAM, ISO 25010:2023.

1 Introducción

1.1 Motivación

En el contexto de la 33ª Feria de Software organizada por la Universidad Técnica Federico Santa María se desarrolló el proyecto de software RVivum, con el objetivo de permitir a las personas explorar patrimonios culturales a través de realidad virtual, en casos donde ya no están disponibles para ser visitados en persona, ya sea debido al abandono, desastres naturales o deterioro de estos lugares.

Las limitaciones que enfrenta la restauración física de espacios patrimoniales en mal estado evidencian la necesidad de soluciones innovadoras y digitales que preserven la memoria cultural del país. Es a partir de esto que RVivum reconstruye digitalmente estos patrimonios culturales, utilizando herramientas de inteligencia artificial generativa y modelamiento 3D, aportando así a la preservación de estos lugares. Esto les da a las personas la posibilidad de visitar otras épocas, conocer sobre la cultura histórica del país y aprender de manera inmersiva y digital.

La problemática que aborda este proyecto de software es la pérdida de acceso al patrimonio cultural destruido, deteriorado o inaccesible en Chile, lo que limita su preservación, difusión y valoración pública. Este problema surge porque gran cantidad de patrimonios ya no pueden ser visitados físicamente, se encuentra en riesgo de desaparecer o no cuenta con mecanismos de mediación cultural.



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título Tesis de Postgrado

Título del trabajo: Análisis de la calidad de proyecto de software RVivum bajo el marco de la norma ISO 25010:2023

Nombre del candidato(a): Vicente Ernesto Gaete Olmos

Carrera / Grado: Ingeniería Civil Informática

Campus: San Joaquín Departamento: Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Marcello Visconti, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

6 meses 12 meses 2 años 3 años 5 años 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 27-05-2026 Firma: 

Estudiante o Candidato(a):

Fecha: 27-05-2026 Firma: 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Las causas pueden ser deterioro por el tiempo, incendios, desastres naturales o el abandono de estos sitios relevantes de parte de instituciones culturales. Por otro lado, nos centramos en lugares donde la reconstrucción física es una solución poco viable para preservar estos sitios.

Los efectos de este problema son la pérdida progresiva de la memoria histórica y colectiva asociada a estos sitios destruidos o inaccesibles. Además, se encuentra la dificultad para transmitir conocimiento patrimonial de forma atractiva y la pérdida de oportunidades educativas basados en experiencias culturales innovadoras. En la **Figura 1** se presenta el diagrama que ilustra el árbol del problema.

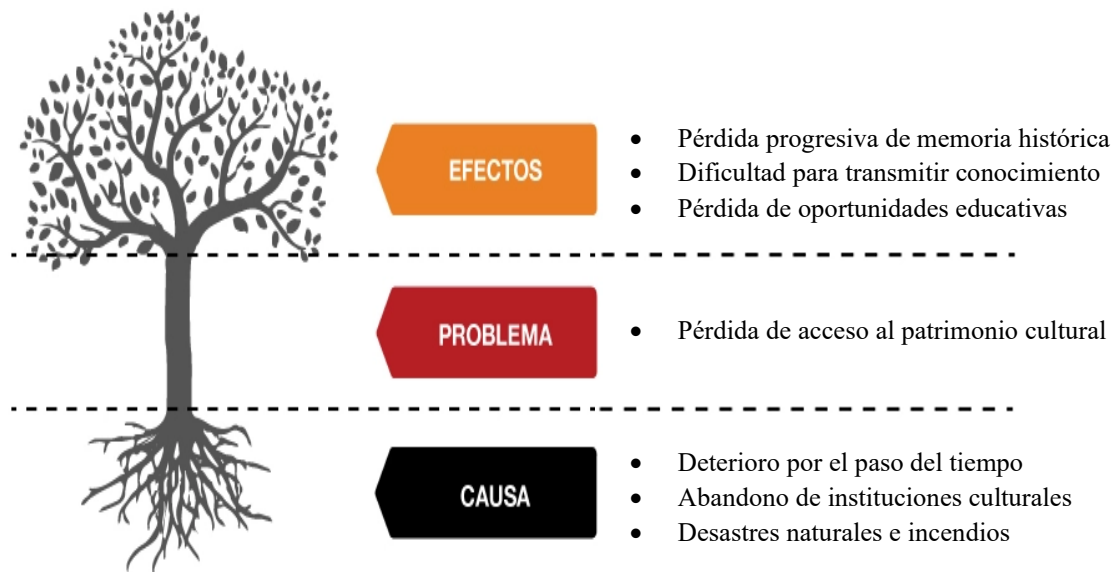


Figura 1. Árbol del Problema de proyecto RVivum

Fuente: Elaboración propia

Así, este proyecto aborda estas causas creando reconstrucciones digitales inmersivas y explorables de patrimonio perdido o deteriorado, permitiendo su preservación, acceso y uso educativo mediante inteligencia artificial y realidad virtual.

Sin embargo, para que esta solución cumpla efectivamente su propósito de preservación y exploración cultural, es fundamental asegurar que el sistema alcance niveles adecuados de calidad. La naturaleza de las aplicaciones de realidad virtual inmersivas e interactivas exigen que el software sea estable, eficiente en el uso de sus recursos y fácil de usar. Esto garantiza que la aplicación responda correctamente tanto a los requerimientos técnicos de su arquitectura como a los requerimientos funcionales de las partes interesadas.

En este contexto, la calidad de software se convierte en un punto fundamental que resulta en la viabilidad del proyecto. Un sistema que presente fallos arquitectónicos o que no cumpla con características críticas, como la mantenibilidad y el rendimiento, podría frustrar la experiencia de usuario y comprometer el valor educativo de la propuesta.

A partir de la problemática señalada, el software debe cumplir con estándares de calidad que aseguren un producto adecuado, confiable, y que resulte en la satisfacción de los clientes. Para esto, se realizará una evaluación integral de la calidad, primero con un análisis de las decisiones arquitectónicas mediante ATAM, seguido de una evaluación sobre los atributos de calidad basados en la norma ISO 25010:2023. Finalmente, a partir de los resultados obtenidos, se propone un plan de mejora para el software.



1.2 Objetivos

El objetivo de esta memoria es proponer un plan de mejora de la calidad para el proyecto de software RVivum, desarrollado en el contexto de la asignatura de Feria de Software, a partir de los resultados de la evaluación arquitectónica mediante ATAM y de la evaluación de la calidad del software basado en el modelo de calidad ISO 25010:2023.

Los objetivos específicos son:

- Identificar y priorizar los atributos de calidad relevantes para este software.
- Evaluar la arquitectura del software mediante la metodología ATAM.
- Evaluar la calidad del software basado en el modelo de calidad de producto ISO 25010:2023.
- Elaborar una propuesta de mejora de la calidad basada en los resultados del análisis arquitectónico y de calidad.

1.3 Estructura del informe

1. **Introducción:** En esta sección se presenta el contexto del proyecto RVivum, la motivación del estudio y el problema abordado. Además, se presenta el objetivo general y los objetivos específicos de este trabajo.
2. **Marco Teórico:** Se definen los conceptos fundamentales para el estudio. Se describe la evaluación arquitectónica mediante ATAM y el modelo de calidad de producto software ISO 25010:2023.
3. **Priorización de atributos de calidad:** En esta sección se describen los atributos de calidad más relevantes para RVivum y se les otorga niveles de prioridad.
4. **Aplicación de ATAM:** Se explica la metodología utilizada para el análisis, se presentan las decisiones arquitectónicas importantes para construir los escenarios de calidad y se identifican los tradeoffs del sistema.
5. **Evaluación de calidad basada en ISO:** En esta sección se presentan las métricas utilizadas para la evaluación de los atributos de calidad seleccionados. Luego se presentan los resultados y el análisis obtenido de las evaluaciones.
6. **Propuesta de mejora:** En este capítulo se proponen mejoras para el sistema, basado en el análisis de los resultados de la evaluación realizada en los capítulos 4 y 5.
7. **Conclusiones:** Se finaliza con las conclusiones obtenidas del trabajo realizado.

2 Marco Teórico

2.1 Conceptos fundamentales

En esta sección se presentan los conceptos relevantes para el análisis desarrollado en esta memoria, estableciendo una base teórica para comprender los criterios utilizados en la evaluación de la calidad del sistema. En primer lugar, el **software** se define como el conjunto de programas, procedimientos y posiblemente documentación y datos asociados que permiten que un sistema informático realice tareas específicas y funcione adecuadamente [1]. En este sentido, el software no se limita únicamente al código fuente, sino que incluye todos los recursos necesarios para su correcto funcionamiento y mantenimiento.

En relación con lo anterior, la **calidad de software** corresponde al grado en que un sistema, componente o proceso satisface los requisitos especificados y las expectativas del usuario o cliente [1]. La calidad es una condición evaluable mediante criterios y métricas previamente definidas, permitiendo analizar distintas características del sistema y detectar oportunidades de mejora.

Un **atributo de calidad** es una característica o propiedad de un software que afecta su capacidad para cumplir con los requisitos del usuario [1]. Estos atributos, tales como la mantenibilidad, el rendimiento o la usabilidad,



permiten descomponer el concepto de calidad en dimensiones específicas del sistema, facilitando un análisis más estructurado y enfocado.

Para que el análisis de los atributos de calidad sea comparable y objetivo se requiere de una **métrica de calidad**, que corresponde a una medida cuantitativa que se utiliza para evaluar un atributo de calidad del software, reflejando el grado en que el software posee esa característica [1]. Las métricas facilitan la comparación de resultados entre distintas evaluaciones, lo que resulta relevante para observar y analizar los comportamientos del sistema.

Finalmente, la **arquitectura de software** es la estructura fundamental de un sistema de software, compuesta por sus componentes, las relaciones entre ellos y los principios que guían su diseño y evolución [1]. Una arquitectura bien definida permite anticipar riesgos y asegurar que el sistema cumpla con los requisitos establecidos, manteniendo un nivel de calidad adecuado frente a distintos escenarios de operación. A partir de esto, las decisiones arquitectónicas influyen en el cumplimiento de los atributos de calidad y en el resultado de las evaluaciones realizadas.

2.2 ATAM

El *Architecture TradeOff Analysis Method* (ATAM) es una metodología de evaluación de arquitecturas de software orientada a analizar cómo las decisiones arquitectónicas influyen en el cumplimiento de los atributos de calidad de un sistema. Fue desarrollada por el *Software Engineering Institute* (SEI) para proporcionar un marco formal que permita identificar riesgos arquitectónicos, puntos sensibles y compromisos entre distintos atributos de calidad.

ATAM se basa en el análisis de escenarios que representan situaciones reales o potenciales de uso, cambio o crecimiento del sistema. A través de estos escenarios, la metodología permite evaluar el impacto de la arquitectura en atributos como desempeño, funcionalidad, disponibilidad y mantenibilidad, facilitando la identificación de decisiones arquitectónicas críticas que pueden afectar el comportamiento del software.

El método utiliza un enfoque basado en escenarios de calidad. Un escenario describe una situación específica que el sistema debe enfrentar y se estructura generalmente en seis elementos: ambiente de operación, estímulo del sistema, respuesta del sistema y medida de la respuesta. Esta manera permite transformar atributos abstractos en condiciones observables y evaluables para facilitar el análisis objetivo y preciso.

El proceso se desarrolla en una secuencia organizada de etapas que buscan alinear los objetivos del negocio con las decisiones técnicas de la arquitectura. Entre las etapas principales se encuentran:

- Presentación del método y del sistema a evaluar.
- Identificación de los drivers del negocio y restricciones.
- Presentación de la arquitectura propuesta.
- Identificación y priorización de atributos de calidad relevantes.
- Generación y análisis de escenarios de calidad.
- Identificación de riesgos, puntos sensibles, puntos de compensación (tradeoffs) y no riesgos.

Un **riesgo** es una decisión arquitectónica potencialmente problemática [2] y que podría comprometer el cumplimiento de un atributo de calidad. Un **punto sensible** es una propiedad de uno o más componentes que es crítica para cumplir con un atributo de calidad específico [2]. Un **tradeoff** es una propiedad que afecta a más de un atributo y es un punto sensible para más de un atributo de calidad [2] y se produce cuando una decisión arquitectónica favorece un atributo en compensación de otro, evidenciando requisitos del sistema potencialmente conflictivos.

Los principales aportes de esta metodología son hacer explícitas las relaciones entre decisiones arquitectónicas, es decir, cómo se compensan entre sí para favorecer los objetivos de calidad del sistema, evidenciando cómo una arquitectura satisface este objetivo de calidad particular. Esto permite fundamentar la toma de decisiones y acciones de mejora usando como base un análisis objetivo.

ATAM se caracteriza por ser una metodología estructurada y sistemática, que promueve la participación de las distintas partes interesadas en el sistema, tales como desarrolladores, stakeholders y usuarios clave. Como resultado del proceso de evaluación, se obtiene un conjunto de riesgos, decisiones sensibles y recomendaciones que sirven de base para la mejora de la arquitectura y, en consecuencia, de la calidad del software.

En el contexto de esta memoria, ATAM se utiliza como una herramienta para evaluar la arquitectura del software RVivum, permitiendo analizar de manera fundamentada su impacto en los atributos de calidad relevantes y apoyar la formulación de una propuesta de mejora.

2.3 Familia ISO 25000 (SQuaRE)

La *International Organization for Standardization* (ISO) es una organización internacional independiente y no gubernamental que reúne expertos de todo el mundo para acordar la mejor manera de hacer las cosas para todo tipo de procesos [3]. Está dedicada al desarrollo de estándares que establecen buenas prácticas para procesos, productos y servicios a nivel mundial. En el ámbito de la ingeniería de software, la ISO, junto con la *International Electrotechnical Commission* (IEC), han desarrollado una serie de estándares orientados a la calidad de sistemas y productos de software.

Dentro de este conjunto de estándares se encuentra la ISO 25000, conocida como SQuaRE (*System and Software Quality Requirements and Evaluation*). Esta es una familia de normas que tiene como objetivo la creación de un marco de trabajo común para evaluar los atributos de calidad de un software [4]. Permite a desarrolladores, evaluadores y organizaciones contar con criterios objetivos que analicen atributos como funcionalidad, eficiencia, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad, teniendo en cuenta las necesidades del usuario y las expectativas del negocio. Para esto, la norma se organiza en distintas divisiones, enfocadas en un aspecto específico del aseguramiento de la calidad.

La ISO 25000 se organiza en cinco divisiones principales, como se muestra en la **Figura 2**.



Figura 2. Divisiones de ISO 25000

Fuente: ISO 25000. (2024). Normas ISO 25000. [4]

- **ISO 2500n – División de Gestión de Calidad:** Define modelos, términos y conceptos comunes a los que hacen referencia las demás normas de la familia SQuaRE. Además, proporciona directrices para la gestión de requisitos y evaluaciones de calidad de los productos de software.

- **ISO 2501n – División de Modelo de Calidad:** Presenta modelos de calidad para productos de software, sistemas, datos, servicios de TI y calidad en uso.
- **ISO 2502n – División de Medición de Calidad:** Establece un marco de medición y define métricas para evaluar propiedades internas, externas y en uso del software.
- **ISO 2503n – División de Requisitos de Calidad:** Ayuda a la especificación de requisitos de calidad basados en modelos y métricas, tanto para desarrollo como para evaluación.
- **ISO 2504n – División de Evaluación de Calidad:** Proporciona recomendaciones, directrices y requisitos para la evaluación sistemática de la calidad de productos de software.

La familia de normas ISO 25000 constituye el marco normativo para el análisis de la calidad del software RVivum, permitiendo estructurar la evaluación de sus atributos de calidad y fundamentar la propuesta de mejora sobre la base de estándares internacionales.

2.3.1 Modelo de Calidad de Producto ISO 25010:2023

El modelo de calidad definido en la ISO 25010:2023 forma parte de la ISO 2501n División de Modelo de Calidad y establece un marco de referencia para evaluar la calidad de productos de software y sistemas informáticos. Este modelo reemplaza y amplía al modelo definido en la norma ISO/IEC 9126, incorporando una visión más completa y actualizada de los atributos de calidad relevantes en sistemas de software modernos.

En 2023, la ISO rediseñó este modelo, por lo que ahora ISO 25010:2023 se enfoca de la Calidad del Producto, es decir, en las características internas y externas del software y aquellas propiedades inherentes al sistema que pueden evaluarse directamente mediante análisis, mediciones o pruebas.

En este ámbito, el modelo se compone de nueve características de calidad: Adecuación Funcional, Eficiencia de Desempeño, Compatibilidad, Capacidad de Interacción, Fiabilidad, Seguridad, Mantenibilidad, Flexibilidad y Protección. Cada una de estas características se subdivide en subcaracterísticas que permiten una evaluación más detallada y estructurada del sistema, representadas en la **Figura 3**. Esta descomposición facilita la especificación de requisitos, la definición de métricas y la evaluación sistemática de la calidad.

CALIDAD DEL PRODUCTO SOFTWARE								
ADECUACIÓN FUNCIONAL	EFICIENCIA DE DESEMPEÑO	COMPATIBILIDAD	CAPACIDAD DE INTERACCIÓN	FIABILIDAD	SEGURIDAD	MANTENIBILIDAD	FLEXIBILIDAD	PROTECCIÓN
COMPLETITUD FUNCIONAL	COMPORTAMIENTO TEMPORAL	COEXISTENCIA	RECONOCIBILIDAD DE ADECUACIÓN	AUSENCIA DE FALLOS	CONFIDENCIALIDAD	MODULARIDAD	ADAPTABILIDAD	RESTRICCIÓN OPERATIVA
CORRECCIÓN FUNCIONAL	UTILIZACIÓN DE RECURSOS	INTEROPERABILIDAD	APRENDIZABILIDAD	DISPONIBILIDAD	INTEGRIDAD	REUSABILIDAD	ESCALABILIDAD	IDENTIFICACIÓN DE RIESGOS
PERTINENCIA FUNCIONAL	CAPACIDAD		OPERABILIDAD	TOLERANCIA A FALLOS	NO-REPUDIO	ANALIZABILIDAD	INSTALABILIDAD	PROTECCIÓN ANTE FALLOS
			PROTECCIÓN FRENTE A ERRORES DE USUARIO	RECUPERABILIDAD	RESPONSABILIDAD	CAPACIDAD DE SER MODIFICADO	REEMPLAZABILIDAD	ADVERTENCIA DE PELIGRO
			INVOLUCRACIÓN DEL USUARIO		AUTENTICIDAD	CAPACIDAD DE SER PROBADO		INTEGRACIÓN SEGURA
			INCLUSIVIDAD		RESISTENCIA			
			ASISTENCIA AL USUARIO					
			AUTO-DESCRIPTIVIDAD					

Figura 3. Características ISO 25010:2023

Fuente: ISO 25000. (2024). ISO/IEC 25010. [5]



A continuación, se describen brevemente las características que conforman este modelo bajo la actualización 2023:

- **Adecuación funcional:** Capacidad de un producto para proporcionar funciones que satisfacen las necesidades declaradas e implícitas de los usuarios cuando se utiliza bajo condiciones específicas [6].
- **Eficiencia de desempeño:** Capacidad de un producto para ejecutar sus funciones dentro de parámetros especificados de tiempo y rendimiento y utilizar los recursos de manera eficiente bajo condiciones determinadas [6].
- **Compatibilidad:** Capacidad de un producto para intercambiar información con otros productos y/o realizar sus funciones requeridas mientras comparte el mismo entorno y recursos comunes [6].
- **Capacidad de interacción:** Capacidad de un producto para ser interactuado por usuarios específicos con el fin de intercambiar información entre el usuario y el sistema a través de la interfaz de usuario para completar la tarea prevista [6].
- **Fiabilidad:** Capacidad de un producto para realizar funciones especificadas bajo condiciones determinadas durante un período de tiempo definido sin interrupciones ni fallos [6].
- **Seguridad:** Capacidad de un producto para proteger la información y los datos de modo que las personas u otros productos tengan el grado de acceso a los datos apropiado según sus tipos y niveles de autorización, y para defenderse frente a patrones de ataque de actores maliciosos [6].
- **Mantenibilidad:** Capacidad de un producto para ser modificado por los mantenedores previstos de manera eficaz y eficiente [6].
- **Flexibilidad:** Capacidad de un producto para adaptarse a cambios en sus requisitos, en los contextos de uso o en su entorno de sistema [6].
- **Protección:** Capacidad de un producto, bajo condiciones definidas, para evitar un estado en el que la vida humana, la salud, la propiedad o el medio ambiente se vean en peligro [6].

3 Priorización de atributos de calidad

3.1 Descripción técnica del sistema

Rvivum es un sistema de software orientado a la exploración de patrimonio cultural mediante el uso de tecnologías de realidad virtual, cuyos modelos son reconstruidos mediante herramientas de inteligencia artificial generativa. Su objetivo es permitir a los usuarios recorrer e interactuar virtualmente con espacios patrimoniales que se encuentran deteriorados, destruidos o inexistentes en la actualidad, ofreciendo una experiencia inmersiva que facilite su preservación, difusión y comprensión histórica.

El sistema está implementado como una aplicación interactiva que integra modelos tridimensionales reconstruidos, información histórica y mecanismos de navegación en entornos virtuales. RVivum está dirigido principalmente a instituciones culturales, como museos y organizaciones dedicadas a la preservación patrimonial, así como a usuarios finales interesados en la exploración de patrimonio digital mediante dispositivos de realidad virtual. En este contexto de uso, el sistema permite a los usuarios seleccionar un espacio patrimonial y recorrerlo mediante interacción directa con el entorno virtual.

Desde el punto de vista tecnológico, RVivum se desarrolla sobre el motor de videojuegos Unity 6, incorporando soporte para dispositivos de realidad virtual mediante el estándar OpenXR y orientado exclusivamente a visores

Meta Quest. Para la gestión de las interacciones dentro del entorno inmersivos se utiliza el Meta XR Interaction SDK, el cual proporciona un conjunto de herramientas para la implementación de funcionalidades orientadas al usuario, tales como la interacción de interfaces, manipulación de objetos, selección de elementos y locomoción virtual. Las funcionalidades personalizadas del sistema, como la lógica de comportamiento e interacción, se implementan mediante scripts desarrollados en C#, que corresponde al lenguaje de programación base soportado para Unity. La aplicación es de tipo standalone, es decir, se ejecuta de manera completamente local en el visor, sin dependencia de servicios externos ni infraestructura backend, manteniendo de forma autocontenida los recursos gráficos, modelos tridimensionales e información patrimonial.

La arquitectura del sistema sigue un enfoque modular, separando los componentes de visualización, gestión de contenido y control de interacción, permitiendo aislar las funcionalidades principales del sistema, facilitando la mantenibilidad y la escalabilidad para la incorporación futura de nuevos espacios patrimoniales o funcionalidades adicionales. Por otro lado, RVivum se estructura en cuatro escenas principales: una escena inicial de tipo lobby que permite seleccionar el patrimonio que desea explorar, y tres escenas patrimoniales independientes, cada una de las cuales representa un entorno reconstruido digitalmente. Todas las escenas comparten una estructura base común, que incluye el sistema de cámara del usuario (Camera Rig), el sistema de locomoción, los componentes de interacción con el entorno y los componentes de interfaz de usuario.

La generación de los entornos digitales y modelos tridimensionales que componen los escenarios patrimoniales fueron creados mediante un flujo de reconstrucción basado en herramientas de inteligencia artificial generativa. En este proceso se emplean plataformas especializadas de generación de modelos 3D de alta calidad a partir de descripciones de texto, imágenes y bocetos, permitiendo crear rápidamente activos 3D, texturizarlos y exportarlos eficientemente, de entre las que se usaron se encuentran Nanobanana, Tripo y Hunyuan. Posteriormente, los modelos generados se procesan y optimizan mediante Blender, donde se realizan reducción de polígonos, optimización del peso de las mallas y ajustar texturas. Este proceso de optimización resulta fundamental para garantizar un rendimiento adecuado dentro de dispositivos de realidad virtual donde los recursos computacionales son limitados. El flujo de generación de los escenarios de RVivum se ilustra en la **Figura 4**.

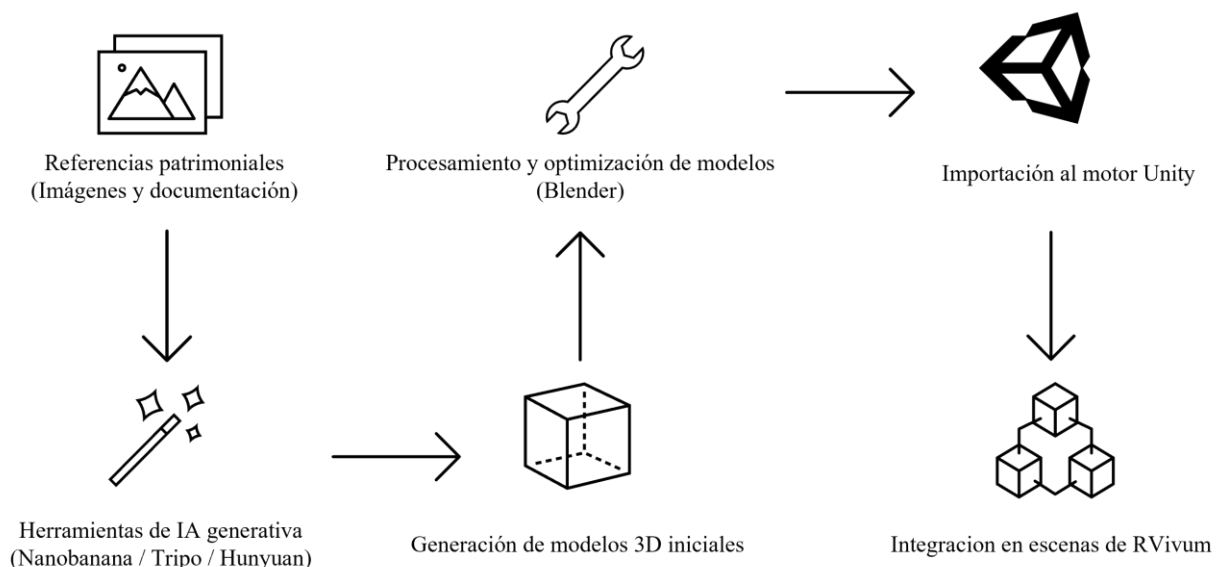


Figura 4. Diagrama de flujo de generación de escenarios

Fuente: Elaboración propia

El flujo de interacción dentro del sistema comienza cuando el usuario inicia la aplicación desde el visor de realidad virtual y accede a la escena inicial. En esta escena se presenta un tutorial de uso de los controles del visor dentro del entorno virtual y luego se presenta el lobby que permite seleccionar el espacio patrimonial que se desea explorar. Una vez seleccionado el patrimonio, el sistema carga la escena correspondiente y posiciona al usuario dentro del entorno virtual, donde puede desplazarse mediante mecanismos de locomoción virtual e interactuar con los elementos presentes utilizando los controladores del dispositivo. Las interacciones incluyen cambios de estado del usuario dentro del entorno, que representan acciones como sentarse en una silla o recostarse en una cama, así como la interacción con paneles informativos y la activación de sonido ambiental. Estas acciones del usuario son gestionadas por el sistema de interacción de Meta XR, que traduce las entradas de los controladores en eventos dentro del motor de ejecución de Unity.

En el contexto de la identificación y priorización de atributos de calidad relevantes, esta descripción técnica permite caracterizar las decisiones arquitectónicas, tecnológicas y de implementación que influyen directamente en propiedades como el rendimiento, la usabilidad, la fiabilidad y la mantenibilidad. En particular estos aspectos impactan de manera directa en la definición de las características del sistema que se evalúan.

3.2 Atributos de calidad relevantes

La identificación de atributos de calidad relevantes es una etapa previa fundamental para la evaluación arquitectónica mediante ATAM y para la evaluación de calidad conforme a la ISO 25010:2023. Esta etapa permite determinar qué características del modelo de calidad del producto software son críticas en el contexto específico del sistema RVivum, orientando la posterior construcción de escenarios de calidad y el análisis de riesgos arquitectónicos.

Como referencia conceptual se utiliza el modelo de calidad del producto software definido en la norma ISO 25010:2023 previamente mencionado, el cual establece un conjunto de características que permiten evaluar la calidad de un sistema software. Sin embargo, no todos estos atributos tienen la misma relevancia para todos los tipos de sistemas, por lo que es necesario seleccionar y priorizar aquellos que tienen mayor impacto en los objetivos y el contexto de uso del sistema evaluado.

La selección de atributos se realizó considerando las características técnicas de RVivum, su propósito como aplicación de realidad virtual inmersiva y las restricciones del entorno de ejecución, específicamente el uso de dispositivos de realidad virtual como Meta Quest 2, los cuales poseen limitaciones específicas de rendimiento computacional. Bajo estas condiciones, se identificaron como atributos de calidad prioritarios los siguientes:

3.2.1 Adecuación funcional

Este atributo se refiere al grado en que el software proporciona las funciones necesarias para cumplir con los objetivos para los cuales fue diseñado. En el caso de RVivum, esto implica permitir la exploración de entornos patrimoniales virtuales mediante navegación dentro del espacio tridimensional, interacción con elementos del entorno y acceso a información contextual asociada al patrimonio representado. La correcta implementación de estas funcionalidades constituye la base del valor del sistema y es condición necesaria para que la aplicación cumpla con su propósito.

3.2.2 Eficiencia de desempeño

Este atributo se refiere al desempeño del software en la realización de sus funciones dentro de unos parámetros de tiempo y rendimiento especificados, con uso eficiente de recursos bajo determinadas condiciones. En aplicaciones de realidad virtual el desempeño es un atributo crítico, ya que influye directamente en la fluidez de la experiencia inmersiva y la comodidad del usuario. En visores de realidad virtual autónomos, como los Meta Quest 2, el procesamiento gráfico, la gestión de modelos 3D y la carga de escena deben realizarse de manera eficiente para evitar caídas en la tasa de fotogramas por segundo (FPS), carga excesiva en el uso de

CPU y GPU, y tiempos de respuesta que puedan afectar la experiencia del usuario. Por esta razón, aspectos como la optimización de modelos 3D, la gestión de recursos y la carga eficiente de escenas son elementos relevantes para este atributo.

3.2.3 Capacidad de interacción

Este atributo se refiere a la capacidad del software para que el usuario interactúe con su interfaz de manera efectiva para completar determinadas tareas. La aplicación está orientada tanto a usuarios con experiencia como a usuarios sin experiencia previa en entornos de realidad virtual. Por lo tanto, resulta fundamental que la interacción con el sistema sea comprensible, fácil de aprender y cómoda de usar. Este atributo abarca aspectos como la claridad de las interfaces del entorno virtual, la facilidad de navegación, la intuitividad de las interacciones de los controladores y la capacidad del usuario para comprender rápidamente el funcionamiento del sistema.

3.2.4 Fiabilidad

Este atributo se refiere a la capacidad del software de mantener un nivel de desempeño adecuado y continuo durante su operación. Para entornos de realidad virtual donde la experiencia del usuario depende de la estabilidad del sistema durante la interacción, este atributo resulta especialmente relevante. La presencia de fallos críticos, comportamientos o interrupciones inesperadas puede afectar la continuidad de la exploración. Es fundamental que el sistema sea capaz de gestionar excepciones y mantener una correcta operación durante las sesiones de uso.

3.2.5 Mantenibilidad

Este atributo representa la capacidad del software para ser modificado, corregido o ampliado a lo largo del tiempo, de manera efectiva y eficiente. En el caso de RVivum, la organización modular de los componentes y la separación de responsabilidades entre los sistemas de interacción, visualización y gestión de contenido permiten que el sistema pueda evolucionar sin afectar significativamente su estructura general. Esto resulta relevante considerando que el sistema podría incorporar nuevos patrimonios virtuales o mejorar los modelos actuales en futuras versiones.

3.3 Priorización de atributos relevantes

Una vez identificados los atributos relevantes, se realiza una priorización considerando su impacto en el funcionamiento del sistema y en la experiencia de usuario. Para esto se evalúa el grado en que cada atributo influye en el cumplimiento de los objetivos del sistema y en las restricciones técnicas del entorno de ejecución. En la **Tabla 1** se muestran los atributos relevantes y su prioridad en el cumplimiento de la calidad del sistema.

A partir de esta priorización, eficiencia de desempeño y capacidad de interacción se consideran atributos de prioridad muy debido a su impacto directo en la experiencia inmersiva del usuario, mientras que adecuación funcional y fiabilidad representan atributos de prioridad alta que garantizan el cumplimiento del propósito del sistema. Por último, mantenibilidad se considera un atributo de prioridad media que favorece la evolución futura del software. Esta priorización orientará la construcción de escenarios de calidad y el análisis arquitectónico que se desarrollará en la aplicación de ATAM y para la evaluación de estos atributos bajo la norma ISO.

Atributo de calidad	Nivel de prioridad	Justificación
Adecuación funcional	Alta	Asegura que el sistema cumpla con las funciones principales para la exploración patrimonial
Eficiencia de desempeño	Muy Alta	Impacta directamente en la fluidez de la experiencia inmersiva y la estabilidad del renderizado en el dispositivo de RV
Capacidad de interacción	Muy Alta	Determina la facilidad con que usuarios pueden interactuar y comprender el entorno virtual
Fiabilidad	Alta	Mantiene una correcta operación del sistema, sin interrupciones ni fallos
Mantenibilidad	Media	Permite la evolución del sistema incorporando nuevos entornos patrimoniales y mejorando la calidad de los modelos actuales

Tabla 1. Atributos de calidad relevantes y su nivel de prioridad

Fuente: Elaboración propia

4 Aplicación de ATAM

4.1 Metodología

La evaluación del sistema RVivum se realiza mediante una aplicación simplificada de ATAM, enfocada en la identificación de atributos de calidad relevantes, la construcción de escenarios de calidad y el análisis de riesgos, puntos sensibles y tradeoffs asociados a las decisiones arquitectónicas del sistema. Esta adaptación considera únicamente las etapas principales de la metodología, debido al alcance acotado del proyecto y al número reducido de partes interesadas involucradas en el proyecto.

ATAM está diseñado para evaluaciones arquitectónicas realizadas en contextos de organizaciones con múltiples participantes, incluyendo arquitectos de software, desarrolladores, clientes y otros interesados. En el caso de este proyecto, al tratarse de un análisis realizado en un contexto académico y centrado en un sistema específico, se adopta una versión simplificada del método, que permite mantener el enfoque analítico de ATAM sin requerir la ejecución completa del proceso formal.

En particular, la aplicación adaptada contempla las siguientes actividades principales: la descripción de los enfoques arquitectónicos del sistema, la identificación y priorización de atributos de calidad relevantes, la construcción de escenarios de calidad derivados de dichos atributos y el análisis de estos escenarios con el fin de identificar los riesgos arquitectónicos, puntos sensibles y posibles tradeoffs asociados a las decisiones de diseño.

El análisis se centra en la arquitectura de RVivum en su versión final implementada, considerando los componentes que conforman la aplicación de realidad virtual. No forman parte del alcance de esta evaluación procesos de gestión del proyecto, aspectos organizacionales del desarrollo ni del modelo de negocio. En este contexto, el siguiente paso consiste en identificar los enfoques arquitectónicos del sistema, que representan las decisiones de diseño que influyen en el comportamiento del sistema frente a escenarios de calidad definidos.

4.2 Enfoques arquitectónicos

Estas decisiones corresponden a mecanismos de diseño, organización de componentes y decisiones tecnológicas adoptadas que determinan el comportamiento del software frente a distintos escenarios de calidad. En el caso de RVivum, la arquitectura se basa en un enfoque modular implementado dentro del motor de desarrollo Unity, donde las distintas funcionalidades del sistema se organizan en componentes independientes encargados de la visualización del entorno, la gestión de interacción en realidad virtual y la administración del contenido patrimonial. Esto permite separar responsabilidades dentro del sistema, facilitando la mantenibilidad del software y la incorporación de nuevos entornos patrimoniales sin alterar la estructura general.

Una de las decisiones arquitectónicas principales es la ejecución local en el visor de realidad virtual como standalone. La aplicación se ejecuta completamente dentro del visor Meta Quest 2, sin depender de servidores, backend ni recursos computacionales externos. Como consecuencia, la gestión eficiente de los recursos del dispositivo se vuelve un elemento relevante para mantener un desempeño adecuado durante la experiencia inmersiva.

Otro enfoque relevante es la organización del sistema mediante escenas independientes y excluyentes dentro del motor Unity. Como se mencionó anteriormente, RVivum se compone de varias escenas patrimoniales que representan los distintos entornos virtuales disponibles para exploración, y cada una de estas escenas incorpora los componentes y activos necesarios para su funcionamiento. Durante la ejecución, el sistema mantiene únicamente una escena activa en memoria a la vez, cargando y descargando los recursos asociados según la escena seleccionada por el usuario. Este enfoque contribuye a optimizar el uso de recursos y simplificando la transición entre los distintos entornos de la aplicación.

Adicionalmente, la arquitectura incorpora un mecanismo de gestión de interacción virtual basado en el estándar OpenXR y el uso de Meta XR Interaction SDK. Este conjunto de herramientas permite abstraer la gestión de los dispositivos de entrada, el seguimiento espacial del usuario y las interacciones dentro del entorno virtual, facilitando la implementación de funcionalidades como la locomoción, selección de elementos y manipulación de objetos. Este enfoque contribuye a mejorar la usabilidad del sistema, proporcionando mecanismos de interacción consistentes y comprensibles para el usuario.

Finalmente, se adopta un enfoque orientado a la optimización manual de los modelos, necesario para garantizar un rendimiento adecuado. Como se mencionó anteriormente, los modelos tridimensionales que componen los entornos patrimoniales son previamente optimizados mediante procesos de reducción de complejidad geométrica y optimización de texturas antes de su integración en el motor de desarrollo. Esto permite mantener un desempeño estable durante la exploración.

4.3 Construcción de escenarios de calidad

Con el objetivo de evaluar las decisiones arquitectónicas del sistema, se definieron una serie de escenarios de calidad que permiten evaluar cómo la arquitectura responde ante distintas situaciones de uso. Estos escenarios se construyen considerando los atributos de calidad relevantes previamente identificados. Para cada escenario se describe el atributo de calidad relacionado, el ambiente de operación en el que ocurre, el estímulo que afecta al sistema, y la respuesta esperada del sistema. Luego se identifican y relacionan los riesgos, puntos sensibles y tradeoffs particulares para cada decisión arquitectónica.

4.3.1 Escenario 1: Interacción simultánea con múltiples objetos en una escena

Atributo: Adecuación funcional

Ambiente: Operación normal del sistema durante exploración de la escena

Estímulo: El usuario se encuentra en un sector donde existen múltiples objetos interactivos y realiza acciones de selección, manipulación de objetos o activación de paneles informativos mediante los controladores del visor.

Respuesta: El sistema detecta correctamente las acciones del usuario, ejecutando únicamente el evento correspondiente al objeto seleccionado, sin pérdida ni activación simultánea no deseada de eventos, manteniendo estabilidad y fluidez.

Decisión arquitectónica	Riesgo	Punto sensible	Tradeoff
Uso del sistema de interacción provisto por el Meta XR Interaction SDK	Conflictos en la detección de eventos cuando existen múltiples objetos interactivos simultáneamente	Número de colliders y raycasts activos por escena	Robustez del Interaction SDK vs Control sobre el procesamiento de eventos
Centralización de componentes de Camera Rig, Locomotion y Interaction compartido entre escenas	Una mala configuración del sistema base puede afectar transversalmente todas las escenas	Acoplamiento entre sistema base y objetos interactivos	Reutilización estructural vs Punto único de fallo

Razonamiento: El uso del sistema de interacción provisto por el Meta XR Interaction SDK responde a la necesidad de implementar de forma robusta y estandarizada los mecanismos de interacción propios de los entornos de realidad virtual, tales como la selección de objetos, la manipulación de elementos y la activación de paneles informativos mediante los controladores del visor. Esta decisión permite la integración de funcionalidades complejas sin requerir el desarrollo de un sistema de interacción propio desde cero, aprovechando las herramientas optimizadas específicamente para el hardware del visor Meta Quest. Sin embargo, cuando el usuario se encuentra en sectores de la escena donde existen múltiples objetos interactivos, aumenta el número de raycasts, colliders eventos de interacción procesados simultáneamente, lo que puede generar ambigüedad en la detección de eventos o activaciones no deseadas si varios objetos compiten por la interacción dentro de un mismo espacio. El tradeoff identificado se establece entre la robustez y estandarización que proporciona el SDK oficial y el menor control sobre el procesamiento interno de los eventos de interacción.

Por último, la decisión de estructurar el sistema sobre una base centralizada de componentes principales, como por ejemplo el Camera Rig, los sistemas de Locomotion y los componentes de Interaction, responde a un criterio de coherencia funcional y reutilización de componentes dentro del sistema. Esto permite que el comportamiento del usuario dentro del entorno virtual sea consistente en todas las escenas, facilitando además la incorporación de nuevos espacios patrimoniales sin necesidad de redefinir los mecanismos de navegación e interacción. Sin embargo, esta centralización también implica que cualquier modificación o error en la configuración del sistema base pueda afectar simultáneamente a todas las escenas del proyecto. El tradeoff se establece entre la reutilización estructural de componentes y la existencia de un punto de dependencia común dentro de la arquitectura.

4.3.2 Escenario 2: Exploración libre de una escena patrimonial en el visor de realidad virtual.

Atributo: Eficiencia de desempeño

Ambiente: Operación normal del sistema

Estímulo: La escena contiene modelos 3D de elevada complejidad geométrica, generando un alto número de triángulos procesados por fotograma.

Respuesta: El sistema mantiene una tasa de fotogramas estable igual o superior a 60 FPS, sin generar latencia perceptible ni interrupciones en la experiencia inmersiva.

Decisión arquitectónica	Riesgo	Punto sensible	Tradeoff
Uso de modelos 3D generados externamente y optimizados manualmente	Modelos con exceso de polígonos pueden provocar una caída en la tasa de FPS	Número de triángulos procesados por fotograma	Fidelidad de modelos vs Desempeño
Carga tradicional de escenas (sincrónica)	Picos de carga inicial	Uso de memoria en inicialización de escena	Simplicidad arquitectónica vs Tiempos de carga
Incremento de realismo visual mediante detalle en texturas y densidad de objetos	Sobrecarga en GPU puede provocar latencia o inestabilidad en RV	Número de materiales y fragmentación de mallas	Experiencia visual vs Desempeño

Razonamiento: El uso de modelos 3D generados externamente y optimizados manualmente responde a la necesidad de lograr representaciones patrimoniales con un alto grado de fidelidad geométrica, lo que contribuye a la adecuación funcional del sistema, ya que permite recrear con mayor precisión estructural cada uno de los entornos históricos explorables. Sin embargo, aumenta el número de triángulos por fotograma, lo que puede provocar una reducción en la tasa de FPS, afectando el atributo de eficiencia de desempeño. El tradeoff identificado se establece entre la fidelidad estructural de los modelos y la estabilidad del rendimiento en tiempo real.

La decisión de implementar una carga tradicional de escenas, en donde el hilo principal de ejecución se bloquea hasta que la escena se cargue completamente (sincrónica), en lugar de mecanismos de carga asíncrona o segmentación dinámica, responde a un criterio de simplicidad estructural y reducción de la complejidad arquitectónica, que se traduce en menor número de componentes a gestionar por escena y menor probabilidad de errores asociados a la transición de escenas. Sin embargo, esto implica una carga concentrada de uso de memoria en el momento de inicialización, afectando los tiempos de carga de estas transiciones. El tradeoff por tanto se produce entre la simplicidad arquitectónica y la demora en los tiempos de carga entre escenas.

Finalmente, incrementar el realismo mediante detalles en texturas y densidad de objetos ambientales busca mejorar la experiencia inmersiva, que contribuye a mejorar la experiencia visual del usuario y la percepción de autenticidad del entorno. Sin embargo, al igual que en el uso de los modelos 3D, esto genera una carga en la GPU del visor, disminuyendo la tasa de FPS y el rendimiento general del sistema. El tradeoff se establece entre la experiencia y riqueza visual del entorno y la fluidez de la experiencia inmersiva.

4.3.3 Escenario 3: Uso del sistema por un usuario sin experiencia en RV.

Atributo: Capacidad de interacción

Ambiente: Operación del sistema

Estímulo: Un visitante utiliza RVivum por primera vez y no tiene experiencia previa en RV.

Respuesta: El usuario puede comprender cómo desplazarse, interactuar con el entorno y explorar el patrimonio virtual en un tiempo breve, sin requerir asistencia externa ni instrucciones adicionales.

Decisión arquitectónica	Riesgo	Punto sensible	Tradeoff
Uso del sistema de interacción provisto por el Meta XR Interaction SDK	Usuarios pueden no comprender los mecanismos de interacción	Configuración de ray interactores, colliders y eventos	Estandarización de mecanismos de interacción vs Curva de aprendizaje

Uso de locomoción limitado basado en movimiento controlado	Posible desorientación espacial o dificultad para comprender el desplazamiento	Configuración del sistema de locomoción	Libertad de movimiento vs Facilidad de aprendizaje
Uso de indicadores visuales para identificar objetos interactivos	Saturación visual cuando existen muchos indicadores en la escena	Cantidad y ubicación de elementos visuales interactivos	Claridad de interacción vs Inmersión visual

Razonamiento: El uso del sistema de interacción provisto por el SDK de realidad virtual de Meta responde a la necesidad de implementar mecanismos de interacción robustos, estandarizados y compatibles con el hardware del visor. Esto permite aprovechar las herramientas oficiales diseñadas específicamente para entornos de realidad virtual de dispositivos Meta Quest y detectar las acciones del usuario de manera consistente. Sin embargo, los usuarios que no poseen experiencia previa en realidad virtual pueden requerir un periodo de adaptación para comprender los mecanismos utilizados por el sistema de interacción. El tradeoff se establece entre la estandarización y robustez de los mecanismos de interacción proporcionada por el SDK y la curva de aprendizaje inicial que enfrenta un usuario novato al utilizar por primera vez este sistema.

La decisión de usar el sistema de locomoción simplificado basado en mecanismos de movimiento limitado responde a la necesidad de facilitar la navegación dentro del entorno virtual para usuarios que no poseen experiencia previa en realidad virtual, lo que permite desplazamientos más predecibles y sencillos dentro del entorno y reduciendo la probabilidad de mareos o desorientación espacial. Esto contribuye a la capacidad del sistema para que el usuario pueda aprender su funcionamiento dentro de un tiempo breve. Sin embargo, esta decisión limita el grado de libertad de movimiento disponible para el usuario dentro del entorno virtual, por lo que el tradeoff se establece entre la facilidad de aprendizaje y la libertad de desplazamiento que podría ofrecer un sistema de locomoción más avanzado.

Por último, el uso de indicadores visuales que permiten identificar los objetos interactivos dentro del entorno visual responde a la necesidad de facilitar la comprensión del sistema y el flujo de este para usuarios sin experiencia previa en realidad virtual. Estos indicadores señalan de forma explícita que elementos pueden ser seleccionados o activados, reduciendo la ayuda externa que pueda necesitar el usuario. Sin embargo, la incorporación excesiva de elementos puede generar saturación visual y afectar la inmersión del entorno. El tradeoff se establece entre la claridad de los elementos interactivos disponibles para el usuario y la preservación de la inmersión visual del entorno representado.

4.3.4 Escenario 4: Interrupción de la experiencia por fallo en interacción

Atributo: Fiabilidad

Ambiente: Operación normal del sistema

Estímulo: El usuario realiza una interacción dentro del entorno virtual que interrumpe la continuidad de la exploración.

Respuesta: El sistema maneja la situación evitando el cierre inesperado de la aplicación, permitiendo continuar con la experiencia sin necesidad de reiniciar.

Decisión arquitectónica	Riesgo	Punto sensible	Tradeoff
Uso de scripts de interacción sin control centralizado de estados de usuario (parado/sentado/acostado)	Errores en interacciones pueden provocar comportamientos impredecibles	Lógica de interacción del usuario y transición entre estados	Simplicidad vs Robustez y control del sistema

No incorporación de mecanismos de recuperación ante fallos en ejecución	La ocurrencia de fallos obliga a reiniciar la aplicación, interrumpiendo la continuidad	Manejo de errores y excepciones durante ejecución	Menor complejidad del sistema vs Baja tolerancia a fallos
---	---	---	---

Razonamiento: La implementación de interacciones mediante scripts propios responde a la necesidad de desarrollar funcionalidades personalizadas y complejas en un corto periodo de tiempo, lo permitió incorporar acciones de usuario fácilmente. Sin embargo, al no existir un mecanismo centralizado de gestión de los estados válidos del usuario y las transiciones entre estos, se incrementó la probabilidad de comportamientos impredecibles que conducían a interrupciones en la exploración. El tradeoff se establece entre la simplicidad en la implementación de interacciones y la necesidad de mantener un control robusto sobre el estado del usuario.

La ausencia de mecanismos de recuperación ante fallos implica que el sistema no posee la capacidad de manejar errores o excepciones durante la ejecución. Esto está asociado a la priorización del desarrollo funcional por sobre la incorporación de lógica adicional de manejo de errores, reduciendo la complejidad de implementación. Sin embargo, resta capacidad a tolerancia a fallos del sistema, impactando en la disponibilidad. El tradeoff se establece entre la simplicidad del sistema y la baja tolerancia a fallos.

4.3.5 Escenario 5: Incorporación de un nuevo patrimonio virtual al sistema

Atributo: Mantenibilidad

Ambiente: Evolución del sistema

Estímulo: Se requiere incorporar un nuevo espacio patrimonial para ser explorado dentro de RVivum.

Respuesta: El nuevo patrimonio puede integrarse al sistema con el modelo y sus recursos asociados sin afectar el funcionamiento de las escenas existentes ni requerir modificaciones estructurales significativas en el proyecto.

Decisión arquitectónica	Riesgo	Punto sensible	Tradeoff
Uso de flujo de trabajo externo para la generación y optimización de modelos 3D	Dependencia de herramientas externas y conocimiento individual de desarrollador	Proceso manual de optimización e integración	Rapidez en la generación de modelos vs Control y trazabilidad del proceso de integración
Ausencia de pipeline estandarizado de integración de activos 3D	Inconsistencias técnicas entre patrimonios	Generación y validación del proceso de integración de activos 3D	Flexibilidad en proceso de integración vs Estandarización, trazabilidad y mantenibilidad del sistema

Razonamiento: El uso de un flujo de trabajo externo para la generación y optimización de modelos tridimensionales responde a la necesidad de producir representaciones patrimoniales con un nivel de detalle adecuado dentro de plazos razonables de desarrollo. Esto contribuye a la fidelidad visual de los entornos reconstruidos, pero introduce una dependencia respecto a herramientas externas y al conocimiento individual del desarrollador en procesos manuales de optimización e integración de los activos, generando un tradeoff entre la rapidez en la generación de modelos 3D y el control sobre el proceso de integración y mantenimiento de los activos del sistema.

La ausencia de un pipeline estandarizado de integración de activos es un factor crítico que responde a la necesidad de un trabajo flexible durante el desarrollo y validar en corto periodos de tiempo activos 3D para los escenarios. Esto produce flexibilidad y rapidez en la integración, pero incrementa la variabilidad e

inconsistencia entre escenas, dificultando el mantenimiento a largo plazo. El tradeoff se produce entre la flexibilidad del proceso actual y la trazabilidad y mantenibilidad del sistema

5 Evaluación de calidad basada en ISO 25010:2023

5.1 Definición de métricas de evaluación

Para la realización de la evaluación, se definieron un conjunto de indicadores, métricas y criterios de aceptación, orientados a medir el grado de cumplimiento de los atributos de calidad seleccionados del modelo de calidad de producto ISO 25010:2023. Estas métricas fueron diseñadas considerando el contexto específico del sistema RVivum y se seleccionaron cubriendo solo las características con prioridad Alta o Muy Alta, que corresponde a Adecuación funcional, Eficiencia de desempeño, Capacidad de interacción y Fiabilidad.

En particular, para evaluar la adecuación funcional, se definieron métricas asociadas a la cobertura de las funcionalidades del sistema y a la precisión de las interacciones. La primera permite cuantificar el grado en que el sistema implementa las funcionalidades especificadas en los requisitos del proyecto. La segunda permite verificar la precisión con la que el usuario ejecuta correctamente las acciones dentro del entorno virtual. Se estableció un umbral de 90% y 95% respectivamente para el criterio de aceptación, dado que se busca un nivel alto de cumplimiento en términos de completitud y correcto funcionamiento.

Para la eficiencia de desempeño, se establecieron indicadores relacionados con la estabilidad de la tasa de FPS, la latencia de carga de escenas y el uso de recursos de CPU y GPU. Para la medición de estos indicadores se utiliza OVR Metrics Tool, la cual es una herramienta oficial de Meta para monitorear distintas estadísticas de rendimiento de aplicaciones en dispositivos Meta Quest [7]. Para el indicador de la tasa de FPS se define un criterio de aceptación de ≥ 72 FPS, basado en la documentación de Meta Horizon para desarrolladores, el cual establece este valor como la tasa mínima para aplicaciones interactivas [8]. Por otro lado, la latencia permite evaluar el tiempo requerido para renderizar y cargar un entorno virtual. Se definió un umbral de ≤ 2 segundos, ya que tiempos superiores producen interrupciones más perceptibles y que afectan la continuidad. Por último, se miden los niveles de rendimiento de CPU y GPU en una escala de 0 a 5, donde valores más altos indican mayor carga para el procesamiento. Esto permite identificar posibles cuellos de botella y conocer si la aplicación se encuentra limitada por alguno de estos recursos. Se establece como criterio de aceptación un nivel de ≤ 3 , lo que indica un uso moderado y eficiente, que se traduce en estabilidad de la ejecución, temperatura adecuada del dispositivo y fluidez del entorno virtual.

Para la capacidad de interacción se definieron indicadores orientados a medir la facilidad con la que usuarios pueden comprender y utilizar el sistema sin asistencia. La facilidad de navegación del sistema busca evaluar la capacidad del usuario para entender el funcionamiento del sistema en general. La comprensión de las interacciones busca medir la capacidad para interactuar correctamente con los elementos informativos e interactivos del entorno patrimonial. Se busca con esto conocer que tan intuitivo y usable es la aplicación para usuarios sin experiencia previa en entornos de realidad virtual. Se establece un criterio de aceptación de 90% para ambos casos, con el objetivo de asegurar que la gran mayoría de los usuarios puedan utilizar el sistema de forma efectiva.

Por último, para la fiabilidad se definió como indicador la capacidad del sistema para evitar fallos durante su ejecución. Esto permite cuantificar la estabilidad del sistema en condiciones reales de uso. Se considera como fallo crítico cualquier evento o situación que interrumpa la continuidad de la experiencia y obligue a reiniciar la aplicación. Se establece como criterio de aceptación un valor $\geq 95\%$, con el objetivo de asegurar que en la gran mayoría de las sesiones de uso se desarrollen sin problema y sin comportamientos inesperados.

En la **Tabla 2** se presenta el resumen de las características evaluadas, los indicadores definidos y sus respectivas métricas y criterios de aceptación.

Característica	Indicador	Métrica	Criterio de aceptación
Adecuación funcional	Cobertura de funcionalidades	$\frac{F. implementadas}{F. especificadas} \times 100$	> 90%
	Precisión de interacciones	$\frac{Interacciones exitosas}{Total de intentos} \times 100$	> 95%
Eficiencia de desempeño	Estabilidad de FPS	Tasa de FPS promedio durante una sesión de ejecución	≥ 72 FPS
	Latencia de carga de escenas	Tiempo promedio en segundos de renderizado de escenas	≤ 2 segundos
	Nivel de rendimiento de CPU	Nivel de CPU (0-5) promedio durante una sesión de ejecución	≤ 3
	Nivel de rendimiento de GPU	Nivel de GPU (0-5) promedio durante una sesión de ejecución	≤ 3
Capacidad de interacción	Facilidad de la navegación del sistema	% de usuarios que logran recorrer el entorno patrimonial sin asistencia	$\geq 90\%$
	Comprensión de las interacciones	% de usuarios que logran interactuar con un panel informativo sin asistencia	$\geq 90\%$
Fiabilidad	Capacidad del sistema de evitar fallos	% de sesiones completadas sin fallos críticos	$\geq 95\%$

Tabla 2. Características evaluadas, indicadores y sus respectivas métricas y criterios de aceptación

Fuente: Elaboración propia

5.2 Resultados y análisis de la evaluación

5.2.1 Adecuación funcional

Para el indicador de la cobertura de funcionalidades se obtuvo un valor del 100%, **cumpliendo con el criterio de aceptación** definido del 90%. Este resultado indica que la totalidad de los requisitos especificados por las partes interesadas se encuentran plenamente implementados y operativos en el sistema. En consecuencia, se afirma que el sistema cubre adecuadamente las tareas y objetivos definidos para los usuarios dentro del contexto de uso previsto.

Por otro lado, la evaluación de la precisión de interacciones registró un valor de 90%, por lo tanto, **no cumple con el criterio de aceptación** definido del 95%. A pesar de que la mayoría de las interacciones fueron exitosas, el margen de error del 10% indica la presencia de fallos recurrentes durante las pruebas de usuario al interactuar con algunos elementos del entorno virtual. Estos fallos se identificaron específicamente en las escenas de la Iglesia San Francisco y de la Escuela Normal de Preceptoras, donde se buscaba activar acciones correspondientes a sentarse y acostarse dentro del espacio virtual, pero estas interacciones fallaron en ciertos casos, lo que sugiere posibles colisiones en la detección de los Ray Interactor de los controladores.

Desde la perspectiva de la calidad del producto, esto evidencia que el sistema dispone de las funcionalidades necesarias para cumplir con los objetivos definidos, sin embargo, existen oportunidades de mejora en la corrección funcional de las interacciones implementadas. En particular, se deben revisar los mecanismos de detección de interacciones, en este caso, los componentes relacionados a Ray Interactor y Ray Interactable, para mejorar la consistencia y robustez de la respuesta del sistema frente a las acciones del usuario.

5.2.2 Eficiencia de desempeño

En relación con el indicador de estabilidad de FPS, el sistema registró un valor promedio de 69 FPS, el cual **no cumple con el criterio de aceptación** definido de ≥ 72 FPS. Este resultado indica que se mantiene una tasa de FPS relativamente alta y estable, sin embargo, no alcanza con el mínimo establecido para asegurar una experiencia completamente fluida en el visor de realidad virtual. Específicamente, se obtuvo que en escenas con mayor carga gráfica se reduce levemente el rendimiento esperado, disminuyendo hasta los 60 FPS, posiblemente por el número de elementos renderizados simultáneamente. En aplicaciones de realidad virtual, la tasa de FPS garantiza una experiencia agradable a la vista y una mejor percepción de realismo, en este caso, una baja tasa de FPS puede afectar la experiencia inmersiva e incluso generar incomodidad en los usuarios durante sesiones largas de exploración.

Respecto al indicador de latencia de carga de escenas, se obtuvo un tiempo promedio de 1,83 segundos desde la ejecución del método LoadScene(s) hasta el estado de disponibilidad completa de las escenas, **cumpliendo así con el criterio de aceptación** definido en ≤ 2 segundos. Esto indica que el sistema es capaz de cargar los entornos virtuales en un tiempo adecuado. Desde la perspectiva de la experiencia de usuario, este comportamiento favorece la continuidad de la exploración.

Para la evaluación de los niveles de CPU, se obtuvo un valor de 3, **cumpliendo así con el criterio de aceptación** establecido de ≤ 3 . Este resultado indica que el procesamiento requerido por la aplicación se mantiene dentro de los márgenes adecuados, permitiendo que el sistema operativo del visor ejecute sus funciones sin saturar los recursos disponibles y manejando una carga de trabajo estable. Para aplicaciones de realidad virtual standalone como es RVivum, esto muestra una utilización eficiente del procesador y deja margen para ampliar las funcionalidades ya implementadas.

Por otro lado, el indicador de los niveles de GPU indicó un valor promedio de 4, llegando en algunos casos a 5, por lo que **no cumple con el criterio de aceptación** definido de ≤ 3 . Este resultado indica que la mayor carga computacional del sistema se concentra en el procesamiento gráfico, lo cual es esperable en aplicaciones de realidad virtual donde la renderización de entornos tridimensionales representa una de las tareas más pesadas en términos de utilización de recursos. En el caso de RVivum, este resultado puede estar asociado a la complejidad de los modelos patrimoniales representados, así como el número de polígonos, texturas y efectos visuales implementados para mantener un nivel adecuado de fidelidad visual en la reconstrucción del entorno histórico.

5.2.3 Capacidad de interacción

En relación con el indicador de la facilidad de la navegación del sistema, se obtuvo un resultado de 16 de 17 usuarios que lograron recorrer el entorno patrimonial sin asistencia, lo que corresponde a un 94%, **cumpliendo el criterio de aceptación** definido de $\geq 90\%$. Esto indica que la gran mayoría de los participantes fue capaz de desplazarse dentro del entorno virtual de forma autónoma, comprendiendo adecuadamente los mecanismos de locomoción y exploración del sistema, demostrando un nivel adecuado de claridad de uso para los usuarios, lo cual es un atributo crítico para aplicaciones de realidad virtual.

Por otra parte, el indicador de comprensión de las interacciones registro igualmente un resultado de 16 de 17 usuarios que lograron interactuar con un panel informativo sin asistencia, correspondiente a un 94%,

cumpliendo también con el criterio de aceptación definido de $\geq 90\%$. Este resultado evidencia que tanto los mecanismos como la interfaz de interacción implementados en el sistema son comprendidos por los usuarios. De igual manera, muestra un nivel de usabilidad adecuado de la capacidad de interacción de los elementos del entorno virtual.

5.2.4 Fiabilidad

Para el indicador de estabilidad del sistema se registró que solo 1 de un total de 18 sesiones presentaron fallos que obligaron al reinicio de la aplicación, lo que corresponde a un 94,4% de sesiones exitosas, por lo que **no cumple con el criterio de aceptación** definido de $\geq 95\%$. Se observó que los fallos se produjeron debido a interacciones con elementos virtuales, específicamente asociadas a acciones de sentarse o recostarse dentro del entorno. En estos casos, el usuario no podía retornar con la exploración normal, interrumpiendo la experiencia y requiriendo reiniciar la aplicación para continuar con su uso. Este comportamiento muestra que el sistema es capaz de completar correctamente la mayoría de las sesiones de uso, pero aún existen escenarios donde se compromete la estabilidad operacional, lo cual es un aspecto crítico para la continuidad de la experiencia del usuario.

En la **Tabla 3** se presenta un resumen de los resultados de las evaluaciones.

Característica	Indicador	Criterio de aceptación	Resultado
Adecuación funcional	Cobertura de funcionalidades	$> 90\%$	Cumple
	Precisión de interacciones	$> 95\%$	No cumple
Eficiencia de desempeño	Estabilidad de FPS	≥ 72 FPS	No cumple
	Latencia de carga de escenas	≤ 2 segundos	Cumple
	Nivel de rendimiento de CPU	≤ 3	Cumple
	Nivel de rendimiento de GPU	≤ 3	No cumple
Capacidad de interacción	Facilidad de la navegación del sistema	$\geq 90\%$	Cumple
	Comprensión de las interacciones	$\geq 90\%$	Cumple
Fiabilidad	Capacidad del sistema de evitar fallos durante operación normal	$\geq 95\%$	No cumple

Tabla 3. Resultados de evaluaciones para atributos de calidad

Fuente: Elaboración propia

6 Propuesta de mejora

A partir del análisis arquitectónico realizado mediante la metodología ATAM y la evaluación de la calidad basado en la norma ISO 25010:2023, se identifican y proponen diversas oportunidades de mejora orientadas a fortalecer y optimizar distintas dimensiones del sistema RVivum.

El nivel de impacto de las mejoras propuestas se deriva de los hallazgos detectados en los capítulos previos, buscando que cada propuesta responda a una necesidad técnica o funcional evidenciada en las dimensiones del sistema. Para esto, se proponen mejoras para todas aquellas características identificadas como relevantes.

6.1 Adecuación funcional

Se propone la incorporación de un componente en C# denominado Interaction Controller, la cual actuará como gestor centralizado de estados de las interacciones, para que en función del contexto del usuario y su proximidad a objetos interactivos específicos, decida cuáles Ray Interactors deben estar activos o priorizados en cada momento. Al restringir dinámicamente la activación de raycasts en zonas con alta densidad de elementos, se elimina la ambigüedad en la detección de eventos cuando existen múltiples objetos interactivos simultáneamente y se asegura que la respuesta del sistema coincida correctamente con la intención del usuario.

Esta mejora responde a los problemas identificados tanto en el análisis ATAM, relacionados con la ambigüedad en la detección de eventos en presencia de múltiples objetos interactivos, como en la evaluación ISO, donde se evidenciaron fallos en la precisión de interacciones, particularmente en acciones como sentarse y recostarse.

6.2 Eficiencia de desempeño

Se propone la configuración del Fixed Foveated Rendering (FFR), basado en la configuración para optimización de aplicaciones de realidad virtual, presentada en la documentación de Meta Horizon para desarrolladores. Esta característica permite reducir la resolución de renderizado en las zonas periféricas del campo de visión del usuario, manteniendo un mayor nivel de detalle en la región central [9]. De esta manera, se reduce la carga de procesamiento gráfico, lo que resulta especialmente adecuado para RVivum, donde se observó niveles elevados de uso de GPU. De igual manera, responde directamente al tradeoff identificado entre fidelidad visual y desempeño, optimizando el uso de recursos y aumentando la tasa de FPS sin afectar significativamente la percepción visual del usuario.

Por otro lado, se propone la utilización del sistema de Occlusion Culling, basado en la guía de buenas prácticas para aplicaciones de realidad virtual proporcionado por la documentación de Meta Horizon para desarrolladores. Este corresponde a un mecanismo mediante el cual se evita el renderizado de objetos de la escena que no son visibles desde la perspectiva actual de la cámara del usuario [10], lo que permite reducir el número de triángulos procesados por fotograma, disminuyendo así la carga sobre la GPU. Esta mejora resulta especialmente relevante en entornos con alta densidad de objetos o estructuras con elevada complejidad geométrica, como es el caso de los escenarios patrimoniales modelados en RVivum, contribuyendo a incrementar la estabilidad de los FPS durante la exploración.

Finalmente, se propone reemplazar el mecanismo actual de carga síncrona de escenas por un enfoque de carga asíncrona mediante el método LoadSceneAsync() de Unity. Este método permite cargar una escena en segundo plano sin bloquear el hilo principal de ejecución [11]. A diferencia de la carga síncrona tradicional, este método no detiene la ejecución de la aplicación, lo que resulta fundamental para mantener la fluidez del sistema y evitar congelamientos visibles de la aplicación. Actualmente, al cargar una escena se detiene el flujo principal hasta completar la carga, lo que puede provocar congelamientos visibles al cargar escenarios que contengan modelos complejos y de alto peso.

6.3 Capacidad de interacción

Se propone como oportunidad de mejora la implementación de un sistema de onboarding adaptativo contextual, diseñado para optimizar la aprendizabilidad y la auto-descriptividad del software. Esto consiste en sustituir el tutorial inicial estático por una guía progresiva que entrega instrucciones en tiempo real según el contexto de interacción del usuario, donde instrucciones y ayudas visuales se muestran dinámicamente cuando el usuario requiere ejecutar una acción específica. Para RVivum, este sistema busca reducir la carga cognitiva inicial, evitando que un usuario novato deba memorizar los controles antes de iniciar la exploración.

6.4 Fiabilidad

En relación con los problemas identificados para la estabilidad del sistema, una parte considerable de estos se aborda indirectamente mediante la mejora propuesta para la adecuación funcional, específicamente a través de la implementación de un script controlador de interacciones basado en contexto. Esta solución contribuye a reducir la ocurrencia de errores asociados a detecciones ambiguas de eventos, mitigando así los fallos críticos observados durante las interacciones evaluadas.

Sin embargo, esta medida no garantiza por sí sola que el sistema sea capaz de recuperarse de una excepción una vez ocurra. Por lo tanto, como medida complementaria para fortalecer la robustez del sistema, se propone la implementación de un sistema de gestión de errores y excepciones, basado en el uso de estructuras de control como bloques try-catch y logging en tiempo real. Esto permitirá capturar excepciones durante la ejecución y gestionar su impacto de manera controlada, evitando la terminación anómala de la aplicación y garantizando la continuidad operativa de la experiencia inmersiva tras un error.

6.5 Mantenibilidad

Para mejorar la incorporación de nuevos patrimonios a la aplicación, se identifica como oportunidad de mejora la estandarización del pipeline de integración de activos tridimensionales. Como se menciona en el capítulo 3, los procesos de importación, optimización y configuración de los modelos 3D generados mediante inteligencia artificial se realizan de manera manual. Esta metodología produce leves inconsistencias técnicas entre patrimonios, incrementa la probabilidad de errores en los modelos y genera una dependencia significativa del conocimiento individual del desarrollador.

En este contexto, se propone la definición de un flujo de integración formal, estructurado y replicable, que establezca convenciones técnicas claras y estrictas respecto a la nomenclatura de activos, escalas, orientación de mallas, configuración de materiales, optimización de polígonos y definición de pivotes. Este enfoque permite asegurar el cumplimiento de requisitos y estándares previamente definidos antes de la integración de los activos en el sistema, facilitando la replicación del proceso, reduciendo errores humanos y aumentando la trazabilidad de los recursos, fortaleciendo de igual manera la capacidad del sistema para ser modificado.

7 Conclusiones

El presente trabajo tuvo como objetivo proponer un plan de mejora de la calidad para el software RVivum, a partir de un análisis estructurado basado en la metodología ATAM y en el modelo de calidad ISO 25010:2023. En este sentido, se logró establecer relaciones entre las decisiones arquitectónicas del sistema y su impacto en los atributos de calidad relevantes, permitiendo identificar riesgos, sensibilidades y oportunidades de mejora. Complementariamente, se cuantificó el cumplimiento de distintos atributos de calidad, lo que proporciona evidencia más objetiva sobre el estado actual del software y sus limitaciones relevantes, alineado con el comportamiento interno del sistema y el contexto de uso de este. De esta forma, se logró obtener una visión integral de la calidad del software, combinando análisis cualitativo y cuantitativo para fundamentar las decisiones de mejora propuestas.



Los resultados obtenidos muestran que RVivum cumple con un estándar de calidad que le permite cumplir con su propósito de preservación y exploración patrimonial inmersiva, sin embargo, este presenta limitaciones en atributos clave, como la eficiencia de desempeño y la fiabilidad. En particular, se identifican riesgos asociados al manejo de recursos durante escenas con alta complejidad geométrica, afectando el uso de GPU y CPU. Asimismo, la capacidad de interacción y la mantenibilidad muestran oportunidades de mejora que impactan directamente en la experiencia de usuario y la escalabilidad del proyecto.

En relación con el análisis de estos resultados, las mejoras propuestas tienen el potencial de aumentar la robustez, escalabilidad y experiencia de usuario, contribuyendo a su viabilidad como solución tecnológica para la preservación y difusión del patrimonio cultural. Entre estas mejoras, se destacan mecanismos de optimización del rendimiento, como el uso de Occlusion Culling y FFR para reducir la carga gráfica, junto con la incorporación de estrategias que permitan estandarizar la integración de modelos y activos virtuales. A partir de esto, resulta relevante como trabajo futuro priorizar aquellas mejoras que impactan de manera más crítica en el sistema, en la experiencia del usuario y en la calidad del producto en general.

Este trabajo aporta una aplicación práctica de metodología formal para la evaluación de calidad en proyectos de realidad virtual dentro de un contexto de desarrollo real. Desde el punto de vista de colaborador en el desarrollo de esta aplicación, este trabajo también representa un aprendizaje técnico que profundiza la importancia de la calidad de software, particularmente en conceptos como la evaluación basada en normas internacionales y en metodologías formales. Además, se presentan desafíos asociados a la implementación de mejoras en la optimización y funcionalidad, lo que impacta en una potencial evolución futura del producto.

Agradecimientos

Me gustaría agradecer a mis padres por aconsejarme y apoyarme siempre en todo momento. A mis hermanos, por acompañarme y hacerme reír. Quiero agradecer a la universidad y a todos los profesores que me ayudaron en este proceso. A mi profesor guía, por su orientación y apoyo durante el desarrollo de este trabajo. Por último, agradezco a todos mis amigos que hice durante este periodo, a quienes me acompañaron hasta el final y a los que hicieron ameno los días de estudio.

Referencias

- [1] IEEE. (1990). IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Institute of Electrical and Electronics Engineers.
- [2] Kazman, R., Klein, M., & Clements, P. (2000). ATAM: Method for architecture evaluation (pp. 1-8). Pittsburgh, PA: Carnegie Mellon University, Software Engineering Institute.
- [3] ISO. (s.f.). Sobre nosotros. Recuperado el 4 de diciembre de 2025, de <https://www.iso.org/es/sobre>
- [4] ISO25000. (2024). Normas ISO 25000. Recuperado el 4 de diciembre de 2025, de <https://www.iso25000.com/index.php/normas-iso-25000>
- [5] ISO25000. (2024). ISO/IEC 25010. Recuperado el 4 de diciembre de 2025, de <https://www.iso25000.com/index.php/normas-iso-25000/iso-25010>
- [6] ISO. (2023). ISO/IEC 25010:2023(en). Recuperado el 4 de diciembre de 2025, de <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-2:v1:en>
- [7] Meta Horizon OS Developers. (2024). OVR Metrics Tool Stats Definition Guide. Recuperado el 15 de marzo de 2025, de <https://developers.meta.com/horizon/documentation/unity/ts-ovrstats/>



- [8] Meta Horizon OS Developers. (2024). Testing and performance analysis. Recuperado el 15 de marzo de 2025, de <https://developers.meta.com/horizon/documentation/unity/unity-perf/#performance-targets>
- [9] Meta Horizon OS Developers. (2024). Fixed foveated rendering (FFR). Recuperado el 15 de marzo de 2025, de <https://developers.meta.com/horizon/documentation/unity/os-fixed-foveated-rendering/>
- [10] Unity Documentation. Unity Manual Version Unity 6.3 LTS. (2026). Occlusion culling. Recuperado el 15 de marzo de 2025, de <https://docs.unity3d.com/Manual/OcclusionCulling.html>
- [11] Unity Documentation. Unity Manual Version Unity 6.3 LTS. (2026). SceneManager.LoadSceneAsync. Recuperado el 15 de marzo de 2025, de <https://docs.unity3d.com/6000.3/Documentation/ScriptReference/SceneManagement.SceneManager.LoadSceneAsync.html>