

**Implementación de un Framework para el desarrollo dinámico en entornos de realidad  
virtual inmersiva**

Leonardo Vélez García

Universitaria Agustiniana  
Facultad de Ingenierías  
Programa de Tecnología en Desarrollo de Software  
Bogotá, D.C.  
2020.

**Implementación de un Framework para el desarrollo dinámico en entornos de realidad  
virtual inmersiva**

Leonardo Vélez García

Directora

Martha Yaneth Segura Ruiz

Trabajo de grado para optar al título de Tecnólogo en Desarrollador de Software

Universitaria Agustiniana

Facultad de Ingeniería

Programa de Tecnología en Desarrollo de Software

Bogotá, D.C.

2020

## **Resumen**

En un proyecto de investigación de la universitaria Agustiniiana se determinó que para un software que iría creciendo constantemente lo ideal sería hacer uso de la arquitectura de microservicios modulares, de esta manera cada vez que se agregara una nueva función no interfiriera con el correcto funcionamiento de la principal, dado esto se propuso un Framework que cumple con los requisitos de micro modularidad lo que permite poder enlazar las App. Este software permite crear, editar y reproducir Tours en realidad virtual, se utilizó el motor gráfico Unity que permite unir exitosamente las aplicaciones, de igual manera ese entorno ha sido el ideal para diseñar y desarrollar el Framework y así cumplir los requisitos del mismo obteniendo una aceptación positiva por parte de los usuarios siendo intuitiva, funcional y portable ofreciendo un agregado ideal para el software al que se unirá conformando así entre las dos Apps una aplicación completa y funcional.

*Palabras clave:* Framework, Realidad Virtual, Unity, Tour.

## **Abstract**

In a research project at the Agustiniana University, it was determined that for software that would constantly grow, the ideal would be to make use of the modular microservices architecture, in this way each time a new function was added it would not interfere with the correct functioning of the Mainly, given this, a Framework was proposed that meets the requirements of micro modularity, which allows to link the App. This software allows creating, editing and reproducing Tours in virtual reality, the Unity graphic engine was used that allows successfully joining the applications, In the same way, this environment has been the ideal one to design and develop the Framework and thus meet its requirements, obtaining a positive acceptance by users, being intuitive, functional and portable, offering an ideal addition to the software to which it will join, thus conforming between the two Apps a complete and functional application.

*Keywords:* Framework, Virtual Reality, Unity, Tour.

## Tabla de contenidos

Introducción .....	7
1. Título del proyecto de software.....	8
2. Planeación del proyecto.....	9
2.1. Objetivos del proyecto.....	9
2.1.1. Objetivo general. ....	9
2.1.2. Objetivos específicos. ....	9
2.2. Planteamiento del problema y/o necesidad .....	9
2.3. Alcance del proyecto.....	10
2.4. Metodología de desarrollo de software .....	10
3. Marco teórico y estado del arte .....	12
3.1. Marco teórico.....	12
3.2. Estado del arte.....	14
4. Especificación de requisitos de Software (IEEE 830).....	18
4.1. Perspectiva del producto.....	18
4.2. Funcionalidad del producto .....	18
4.3. Características de los usuarios .....	19
4.4. Restricciones.....	19
4.5. Suposiciones y dependencias.....	20
4.6. Requisitos específicos .....	21
4.6.1. Actores/roles.....	21
4.6.2. Requisitos funcionales. ....	22
4.6.3. Diagrama de casos de uso. ....	23
4.6.4. Especificaciones de los casos de uso. ....	23
4.7. Requisitos de rendimiento .....	29
4.8. Restricciones de diseño .....	29
4.9. Atributos del software del sistema .....	30
5. Diseño del software (ISO -12207-1) .....	32
5.1. Diseño de la arquitectura de software .....	32
5.2. Diseño detallado del software.....	35
5.2.1. Diagrama de clases. ....	35

5.2.2. Diagrama de paquetes.....	36
5.2.3. Diagramas de despliegue.....	37
5.3. Diseño de la interfaz.....	38
5.3.1. Interfaz gráfica de usuario.....	38
Interfaces de entrada.....	48
Interfaces de salida.....	49
6. Implementación.....	52
6.1. Plataformas de desarrollo.....	52
6.2 Base de datos.....	52
6.3. Infraestructura de hardware y redes.....	53
7. Pruebas del software.....	55
7.1. Pruebas del software.....	55
7.2. Pruebas de usabilidad.....	56
Referencias.....	66

## **Introducción**

Diseñar, programar y crear entornos de realidad virtual inmersiva puede llegar a ser complejo dado que dependiendo de la necesidad y lo tedioso de las estructuras se torna frustrante para cumplir un objetivo y hasta puede volverse una larga tarea que puede traer un resultado no deseado o peor aún el abandono del propósito.

Este proyecto de grado se enfoca en el uso de un Framework, estando en un apartado distinto a la aplicación principal donde será implementado, siendo este una buena opción para escalar el proyecto a una estructura de modularidad facilitando el unir la aplicación a un programa principal y sin problemas permitir activar o desactivar el apartado eludiendo el inconveniente de afectar a la funcionalidad del programa madre. En su construcción se utilizó la metodología de Prototipo y realizando las cuatro etapas que la conforman se logra la primera versión del Framework.

La App presenta un entorno que permite al usuario diseñar su propio recorrido dentro de la aplicación en un apartado diseñado especialmente para cumplir con ello, se espera que a partir de un Framework el usuario tenga alcance a los entornos prediseñados o si desea importar objetos o imágenes para la personalización a su gusto, consiguiendo él podrá darle un uso de ruta informativa con la mayor accesibilidad posible. Todo este proceso se podrá realizar gracias al Framework propuesto, supliendo así también la brecha que hay entre acceder a un recorrido virtual y poder crearlo según necesidad y gustos.

### **1. Título del proyecto de software**

Implementación de un Framework para el desarrollo dinámico en entornos de realidad virtual inmersiva.



## **2. Planeación del proyecto**

### **2.1. Objetivos del proyecto**

#### **2.1.1. Objetivo general.**

Desarrollar un Framework para entornos de realidad virtual inmersiva con el fin de permitir al usuario personalizar ambientes de la mejor manera posible.

#### **2.1.2. Objetivos específicos.**

Identificar los patrones para aplicar en el entorno de personalización.

Establecer características y criterios para agregar o modificar objetos.

Diseñar el entorno y sus funciones para que el usuario pueda editar intuitivamente.

Construir el Framework para orientar, apoyar y facilitar la generación de entornos en realidad virtual.

### **2.2. Planteamiento del problema y/o necesidad**

Con los avances tecnológicos ahora es posible interactuar con entornos digitales de manera cada vez más inmersiva, permitiendo a las personas sentir que sus acciones tienen reacciones similares a la vida real o en otros casos de forma más lúdica con la mayor inmersión que un mundo virtual pueda dar. Pero para hacer uno o varios entornos se necesita muchas horas de trabajo, por lo general de un equipo especializado en el tema de modelado, programación, entre otros; a pesar de ello la inmersión se vuelve un tema subjetivo ya que es accesible dependiendo de la condición física o mental del usuario, como ejemplo un sujeto que puede ver como el promedio afortunadamente tendrá una mejor sensación de inmersión que una persona con discapacidad visual ya que él no podrá percibir visualmente los que se muestre en el visor de realidad virtual. A partir de estos y más problemas se han realizado varios proyectos para que la realidad virtual sea lo más accesible.

La universitaria Agustiniense como en cualquier otro entorno de enseñanza, laboral o de entretenimiento puede ser escalada virtualmente y eso hicieron la ingeniera Martha Yaneth Segura Ruiz y su equipo de investigación conformado por un asesor de carácter externo y estudiantes de la universitaria, con el fin de solucionar la pregunta planteada en su proyecto de investigación, basados en argumentos del ministerio de las TIC se cuestiona ¿Cómo debería desarrollarse un software de realidad virtual inmersiva que cumpla con las necesidades de accesibilidad y supere las barreras de acceso? A partir de esta problemática se necesita saber que componentes usar, dónde y cómo implementarlos conformando en esencia la arquitectura de microservicios modulares de tal forma que el software cuente con funciones a parte de la principal para que no afecten su uso y sea

escalable de la mejor manera posible; iniciando con un valor agregado, por esta razón y aportando a la escalabilidad propongo la implementación del Framework en el proyecto, entonces ¿Cómo utilizar Framework para implementar una función que contribuya a la modularidad en una App de realidad virtual inmersiva?

### **2.3. Alcance del proyecto**

Para el alcance de este proyecto se tiene claro el tiempo de desarrollo y los objetivos a cumplir, al ser un proyecto que se pensó para ser parte de un programa modular este también se convierte en una aplicación escalable en cierta medida, teniendo la perspicuidad de qué es lo que se busca realizar en cuanto a sus futuras funcionalidades, esto pone al proyecto en una posición favorable para aumentar sus competencia y así implementar a mayor escala con dispositivos más inmersivos lo que permitiría una edición más óptima o precisa a cara del usuario.

Este proyecto busca ser una herramienta útil para la creación de recorridos para realidad virtual inmersiva permitiendo a los usuarios promedio diseñar, construir y exhibir lo realizado desde dispositivos Android o computadoras con sistema operativo Windows, todo esto complementando al proyecto de investigación RV 2020 nombrado Prototipo App VRI-Agustiniana, proporcionando un aumento en funcionalidades y/o usuarios convirtiéndolo en un programa más amplio frente a productos similares en desarrollo o existentes en el mercado.

El tiempo ha sido un factor clave en el desarrollo de este Framework de edición ya que obligó a limitar los objetivos y las funcionalidades del mismo, pero aun así permitieron concretar lo propuesto en los lapsos de tiempo dejando una base para retomarlo con actualizaciones de funcionalidad u optimización si así se desea.

Aún no es claro el límite que tenga el prototipo, pero si se puede evidenciar que puede contar con una amplitud de mejoras aptas para continuar con su evolución llevando al Framework a competir en un nuevo mercado en crecimiento.

### **2.4. Metodología de desarrollo de software**

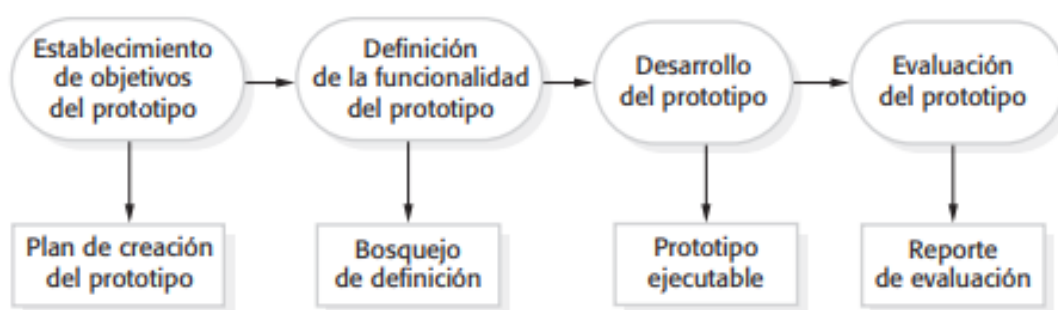
Un prototipo es una versión inicial de un sistema de software que se usa para demostrar conceptos, tratar opciones de diseño y encontrar más sobre problemas y sus posibles soluciones (Somerville, 2011, pág. 45) esta metodología es utilizada en un sinfín de diversas aplicaciones gracias a que se adapta a la mayoría de proyectos ya que es económica y eficaz tanto para los desarrolladores como para los clientes. Cuenta con cuatro etapas que son:

Primera etapa: establecimiento de objetivos del prototipo, es la comunicación que hay con el cliente y el equipo de trabajo que llevará a cabo el desarrollo, esto con el fin de definir los objetivos del producto; En esta etapa se hace el plan de creación del prototipo.

Segunda etapa: definición de la funcionalidad del prototipo, continuando con la comunicación después de determinar qué objetivos se deben alcanzar, el cliente y desarrolladores definen la funcionalidad que tendrá el prototipo. Se obtiene el bosquejo de definición.

Tercera etapa: desarrollo del prototipo, esta etapa es la construcción del prototipo, siguiendo lo acordado en las reuniones con el cliente los desarrolladores construyen el prototipo ejecutable.

Cuarta etapa: evaluación del prototipo, dependiendo del resultado que haya en la evaluación, el cliente decide si es el producto deseado, si no se debe hacer una iteración de los pasos para obtenerlo cambiando o agregando objetivos y funcionalidades, todo esto se hace en un reporte de evaluación.



**Figura 1.** Ciclo de desarrollo prototipo recuperado de Somerville (2011) ingeniería de software.

La metodología prototipo al ser un modelo de desarrollo evolutivo nos permite implementar la escalabilidad y los entornos en realidad virtual inmersiva que se quiere para la aplicación dado que sus etapas son óptimas para llevar a cabo una construcción sólida apuntando a lograr una arquitectura de microservicios modulares, esto hace que prototipo sea la metodología ideal para implementar el Framework en el desarrollo del proyecto de realidad virtual inmersiva de la universitaria Agustiniiana.

### 3. Marco teórico y estado del arte

#### 3.1. Marco teórico

Un Framework puede facilitar y evitar tareas que ya fueron desarrolladas para volver a implementarlas, esto optimiza el tiempo dedicado a un apartado para enfocarlo en otro y así permitir al equipo de desarrollo dedicarse a otros objetivos del proyecto para mejorarlos, optimizarlos, solucionar errores o implementar nuevos. Edwin Mauricio Hincapié Montoya y Cristian Andrés Díaz León de la institución universitaria Salazar y Herrera y de la universidad de Medellín propusieron e implementaron “un Framework para guiar el desarrollo de aplicaciones relacionadas con el patrimonio cultural”, la aplicación “permite a los ciudadanos observar patrimonios arquitectónicos que ya no existen físicamente”. (Hincapié, Díaz, 2014, p. 18).

La utilidad ofrecida por un Framework en principio es muy concreta, pero con el avance tecnológico y su implementación exponencial este tipo de software ha evolucionado para hacer más fácil la programación, al ir acorde con el mismo crecimiento que se vive en la tecnología se presencia el acogimiento del mundo gracias a sus prestaciones, como lo aclaran los ingenieros de sistemas Villalobos, Camacho y Biancha de la universidad tecnológica de Pereira, ellos dicen:

“Un Framework agrega funcionalidad extendida a un lenguaje de programación, automatiza muchos de los patrones de programación para orientarlos a un determinado propósito, proporcionando una estructura al código, mejorándolo y haciéndolo más entendible y sostenible, y permite separar en capas la aplicación. En general, divide la aplicación en tres capas:

La lógica de presentación que administra las interacciones entre el usuario y el software. La Lógica de datos que permite el acceso a un agente de almacenamiento persistente u otros.

La lógica de dominio o de negocio, que manipula los modelos de datos de acuerdo a los comandos recibidos desde la presentación.” (Villalobos, Camacho, Biancha. 2010. p. 178).

La realidad virtual se acerca cada vez más a una total inmersión esto gracias a los avances tecnológicos y la creación del triángulo siendo este la base de la realidad virtual inmersiva que consta de tres características:

“tiempo real, que permite dirigir la dirección hacia donde moverse en el interior del escenario o hacia dónde dirigir la mirada, inmersión completa por el interior del mismo, perdiendo contacto con la realidad del exterior al percibir únicamente los estímulos del mundo virtual; e interacción con los elementos que lo conforman, que permite interactuar con el mundo virtual a través de diversos dispositivos de entrada” (Pérez, 2011, p. 8).

A pesar del acogimiento que ha tenido esta práctica de programación es poco el material que se encuentra diseñado o documentado para videojuegos y más concreto para RV y AR, esto es gracias a que en gran parte se ha aplicado en desarrollos web, Michal Ponder, George Papagiannakis, Tom Molet, Nadia Magnenat-Thalmann, Daniel Thalmann de la universidad de Geneva de Suiza para su proyecto “VHD ++ Development Framework:

Towards Extendible, Component Based VR/AR Simulation Engine

Featuring Advanced Virtual Character Technologies” quisieron entender e implementar un Framework, empezaron por “searching for possible approaches and solutions we looked closer to frameworks and components being presently software engineering concepts of rapidly growing importance” (Ponder, Papagiannakis, Molet, Magnenat, Thalman. 2003 p. 2). Determinaron que “Frameworks offer architectural complexity curbing through predefined, reusable system design, common class vocabulary and uniform conventions. They are “born for” extensions, customisations, and replacements.” (Ponder et al. 2003 p. 2). Su conclusión es una guía para crear el prototipo del Framework para RVI ya que determina este tipo de ajustes o convenciones.

La Arquitectura microservicios modulares es un término reciente pues esta arquitectura surge alrededor de los inicios de esta década, se popularizó para ser la arquitectura de muchos proyectos de software ya que adopta características de la arquitectura monolítica, pero solucionando el problema de modularidad que esta tiene, básicamente esa arquitectura tenía conflictos a la hora de agregar o cambia una función del programa o página web, sumando los cambios constantes de modelo de negocio que traducido al desarrollo se convierten en más requerimientos esto convierte la modularidad en algo insostenible.

Definiendo la arquitectura de microservicios se logra establecer que, para construir una aplicación como un conjunto de servicios, donde cada uno de estos es independiente del otro y hasta pueden ser escritos en lenguajes diferentes y mantenidos por equipos diferentes, son factores cruciales para obtener los resultados deseados (González, 2018, p. 37).

Estas tecnología promete muchos avances y un ejemplo es la aplicación que se le puede dar para simular situaciones del mundo real con el fin de instruir como lo han logrado varios simuladores de vuelo para que los pilotos practiquen o simuladores de combate para entrenar fuerzas militares, hasta se ha implementado la realidad virtual para enseñar a operar maquinaria en la industria petrolera marina como es el caso del simulador ROV multiplayer para escenarios de petróleos submarinos, el simulador ayuda al usuario a realizar con éxito las tareas asignadas permitiéndoles

a los operadores aprender a controlar un ROV (Remotely Operated Vehicle) (Beltrán, Pérez, Ochoa, Ontiveros, 2017, p. 67).

Esperamos que estas herramientas y su implementación sean adecuadas para el contexto y el objetivo del proyecto, intentando llegar a un mayor número de usuarios dejando una base que será el Framework para aplicar futuras funciones en la App o crear nuevas dando así la modularidad deseada.

### **3.2. Estado del arte**

La realidad virtual ha tenido una constante evolución para llegar a ser una tecnología de nivel comercial, su concepto es aplicable en cualquier locación a pesar de que aún hay barreras para una implementación de gran escala esto es gracias a que depende en gran medida del hardware y en el peor de los casos se tiene costoso elevado, con los avances actuales de poder de cómputo podemos acceder a una experiencia de entrada aceptable desde un Smartphone, como ejemplo están los videos en 360° de YouTube, para usar esta tecnología solo se debe contar con una conexión a internet, un Smartphone con giroscopio y un visor donde poner el celular para tener la inmersión. Otro uso es en los simuladores ya sea para entretenimiento o para la instrucción de maquinaria o vehículos de transporte, se puede acceder a él con un computador que cuente con una tarjeta gráfica dedicada y un visor compatible con el sistema.

En Japón un museo de arte digital permite tener una inmersión en diferentes ámbitos digitales y físicos, desde la realidad aumentada hasta la realidad virtual y otros tipos de implementación en arte con tecnología, es un museo sin mapa o recorrido predefinido, se trata de dejar que la curiosidad nos guíe para elegir el camino por el cual comenzar y así conocer las obras que lo componen, obras de arte de instalación digital que forman un mundo sin fronteras que salen de las estancias se comunican con otras obras, a veces se mezclan entre sí e incluso permiten la interacción fomentando la participación de los espectadores (Espada, 2019, p. 82).

Hay proyectos de realidad virtual inmersiva que buscan relatar hechos con la ayuda de esta tecnología, uno de ellos es la recreación del momento en que un inmigrante es abatido por una patrulla estadounidense en la frontera entre México y Estados Unidos (Domínguez, 2015, p. 415). La recreación de esos hechos conlleva tiempo desarrollar en realidad virtual pero así mismo se convierte en otra forma de narra los acontecimientos agregando la inmersión al periodismo.

En Colombia el ministerio de las TIC dotó de tabletas a instituciones educativas públicas después de ello se identificó la necesidad de implementar aplicaciones educativas para ejecutarse

en dichas tabletas, a partir de ahí se propuso construir un Framework que cuente con ciertos aspectos técnicos y pedagógicos para el desarrollo de las aplicaciones formativas, se desarrolló e implementó dando los resultados esperados a pesar de los inconvenientes que se presentaron demostrando así la utilidad ofrecida por dicho Framework (Cuervo, Ballesteros, 2016, p. 128).

El desarrollo en Unity o motores gráficos similares abrió un campo que no se tenía previsto antes como lo es ahora, este programa ha facilitado avances impresionantes y la ciencia no se ha quedado atrás, con investigaciones o experimentos se puede determinar o mirar distintos lados de la moneda, en especial comportamientos humanos implementado tecnología VR o AR, Jack Brookes, Matthew Warburton, Mishary Al Ghadier, Mark Mon-Williams y Faisal Mushtaq de la universidad de Leeds Reino Unido crearon un Framework que les facilitara la construcción de entornos en RV al automáticamente iterar en prácticas de programación que se necesitarán en cada experimento si necesidad de volver a programarlas gracias al Framework, en resumen como ellos mismos explican el Framework sirve para “provides the nuts and bolts that work behind”. (Brookes et al, 2019, p. 457).

En la actualidad existen bibliotecas digitales que permiten acceder a contenido informativo con el que se puede interactuar, este tipo de archivos se encuentran en formatos 3D, para lograrlo se escanea el objeto, luego se realiza su digitalización, y así podrá subirse a la plataforma donde el usuario acceda por medio de internet, interactuar e informarse, en Sudáfrica diseñaron y crearon un entorno en realidad virtual donde el visitante puede observar e interactuar con el entorno de los monumentos o estructuras patrimoniales mediante un visor, teclado y ratón (Wessels, Ruther, Bhurtha, Schroeder, 2014, p. 4) para realizar ese proyecto los desarrolladores crearon su recorrido en Unity demostrando las cualidades con las que cuenta este motor gráfico ya que ellos lograron implementar correctamente texturas de alta calidad en los entornos digitalizados.

Otro ejemplo es VR-Riders como lo llaman los desarrolladores de esta aplicación la cual consiste en unir distintos dispositivos de realidad virtual (Oculus o HTC VIVE) para reproducir e interactuar con imágenes de Google Street View, combinando con narrativas y así dar una experiencia al usuario. (Wang, Ijaz, Calvo, 2017, p. 1).

Los museos en el mundo están optando por la conservación digital de sus colecciones, así como lo han realizado algunas bibliotecas o como fue el caso en Japón para mejorar la experiencia, algunos están realizando esas conversiones mediante un Framework llamado Vir Muf (Virtual Museum Framework) el cual es:

“Is a set of tools that can be used by non-developers to easily create and publish 3D virtual museums in a very short time. This way, Museum staff doing collection digitization can also publish 3D virtual museums to exhibit these collections. VirMuF is open-source; hence, teams including software developers can further extend VirMuF to fit their needs.” (Elfaragy, Rizq, 2018, p. 175).

Los Frameworks han sido de utilidad para investigadores, científicos, funcionarios y público como es el caso de una investigación que presenta “Geospatial VR”, es un Framework de código abierto que sirve para crear entornos en 3D del mundo real para reproducirse en plataformas Web o dispositivos móviles mediante Realidad Virtual a diferencia de otros este puede presentar simulaciones de agua, fuego y daños en el terreno, al igual que otros está diseñado y desarrollado en el motor Unity (Sermet, Demir, 2020, p. 2).

En Texas Tech University, Parviz Safadel y David White decidieron identificar mediante un experimento con realidad virtual inmersiva que beneficios se obtendrían en la enseñanza y el estudio mediante el uso de esta tecnología, los temas que se implementaron fueron ciencias, matemáticas, tecnología e ingeniería, pidieron a los estudiantes de pregrado y posgrado que realizarán el recorrido donde les mostraron maquetas interactivas en 3D con la información correspondiente, al terminar el recorrido debían completar un examen que contenía preguntas relacionadas a la información de cada maqueta organizadas por los temas implementados, con este experimento concluyeron que “VR visualization had a positive compensating impact on students with low spatial ability.” demostrando una vez más que este tipo de tecnología no solo tiene un uso de ocio sino también educativo teniendo en algunos casos mayor impacto que las metodologías tradicionales. (Safadel, White, 2020, p.1.).

El diseño y desarrollo de espacios enfocados a la creación de objetos y entornos para realidad virtual o aumentada se puede llevar a cabo en diversos programas que cuentan con su propio motor gráfico permitiendo editar o crear nuevos objetos virtuales, para crear entornos 3D se necesita conocimiento y horas de trabajo, Álvaro Cid Rodríguez y Dr. Mario Martínez Zarzuela de la universidad de Valladolid desarrollaron un diseñador de espacios 3D para realidad aumentada y virtual inmersiva con el objetivo de facilitar el desarrollo de dichos entornos, quitando de la ecuación el programar y otras actividades que se necesitan para diseñar espacios en tercera dimensión desde una computadora.

Estos son solo ejemplos de lo que se puede hacer con la realidad virtual y la tecnología relacionada a ella, sirven como modelo para este proyecto y así llegar al objetivo final, permitir crear entornos con mayor facilidad y accesibilidad, como una opción de uso que tendrá la



aplicación a parte de su función principal ya desarrollada por el equipo de investigación donde se planteó y solucionó con el programa de realidad virtual inmersiva que ofrece un tour de la universitaria Agustiniiana.

## **4. Especificación de requisitos de Software (IEEE 830)**

### **4.1. Perspectiva del producto**

Con este producto de software se espera lograr como objetivo principal facilitar la creación y edición de tours para realidad virtual inmersiva, permitiendo agrupar objetos e imágenes y así el usuario pueda componer según menester.

Este prototipo se enlazará a un programa principal que funcionará como menú para acceder a la funcionalidad del editor de tours, aparte de estar construidas en el mismo motor gráfico (Unity) se relacionan con la función reproducir un recorrido en realidad virtual que hereda de la aplicación principal, el Framework solo cuenta con esta correlación, esto se debe a que el software App VRI-Agustiniana posee una estructura modular permitiendo a la aplicación heredar dicha competencia, a parte como mayor ventaja puede funcionar acorde sin afectar el funcionamiento de la aplicación principal, de esta manera se cumple en cierta parte con el objetivo de la aplicación padre, ser un software que cuente con servicios modulares, el editor se puede desenlazar cuando se le desee teniendo el plus de acatar a cualquier necesidad ya sea solo reproducir un Tour o crear/editar.

### **4.2. Funcionalidad del producto**

Las funcionalidades del editor de tours empiezan con un menú donde el usuario elige entre las funciones de crear y cargar.

Para crear, el usuario en primera instancia digita el nombre del proyecto que desee, después anotará una descripción del tour (no obligatorio).

Para cargar, al usuario se le mostrará un panel donde están los tours guardados, organizados por slots junto con el nombre del tour que haya creado anteriormente (máximo 4 espacios de guardado) allí él podrá elegir qué tour desea cargar.

Una vez creado o cargado el Framework generará y mostrará el entorno. En este apartado del programa el usuario tiene al alcance varias funciones, las cuales son: agregar una escena (máximo 10 escenas por proyecto), moverse entre escenas con un botón donde se despliegan las respectivas escenas creadas por el usuario, estas están enumeradas hasta diez, también puede agregar un punto de control que sirve para desplazarse entre escenas cuando esté en modo reproducción, agregar objetos 3D para adecuar el tour al gusto y así exhibir lo deseado para ello se mostrará un panel donde el usuario elegirá qué objeto desea importar, se cuenta con el botón de guardado que almacenará en el slot que desee el usuario todo lo realizado en el modo editor, también está el botón para cambiar de imagen 360° allí se despliega un panel donde el usuario elige qué imagen desea

importar de su dispositivo, además se cuenta un botón de reproducción que activa y desactiva el modo de realidad virtual para interactuar con los puntos de control permitiendo observar las escenas y los objetos que estén en ellas, y para finalizar está el botón de salir que devuelve al usuario al menú de elección donde podrá elegir nuevamente entre crear o cargar.

Todas las funciones se enlazan con el guardado y cargado ya que estas dos se encargan de administrar lo realizado por el usuario, la mayoría de las acciones no tienen un orden para realizarse dando así campo libre para la imaginación del usuario.

#### **4.3. Características de los usuarios**

Los usuarios de este editor de tours en realidad virtual tienen el siguiente perfil:

Edad aproximada: 18 años a 50 años para usar el modo editor y desde los 12 años para reproducir las escenas.

Género: Sin restricción.

Conocimiento tecnológico: uso promedio de una aplicación móvil. (Redes sociales, editor de fotos etc.).

Educación: Superior para la creación básica o reproducción, experto en modelado 3D para importar adecuadamente un modelo en la escena.

#### **4.4. Restricciones**

Para la realidad virtual se encuentran varias limitaciones en primer lugar el soporte bidimensional identificados en los límites físicos que se traspasan y se tornan “reales”, un ejemplo es siendo espectadores de un dibujo, sin darnos cuenta, tendemos a prolongar un trazo horizontal en un papel como si éste fuera más ancho que el papel mismo y describiera un espacio implícito.

Una limitación más técnica son las del hardware y software que dependiendo de su potencia y optimización pueden ofrecer una mejor experiencia de realidad al usuario ofreciendo una representación idónea del espacio cartesiano y de la geometría euclídea (espacio geométrico que hace referencia a los axiomas de Euclides), aun siendo creíble para el cerebro humano se debe tener claro que el uso de estas tecnologías no es más que de entretenimiento, simulación o educación diferenciándolo totalmente de la realidad ya que actualmente no se puede comparar con una sensación de realidad en ninguno de los ámbitos ofrecidos por esta tecnología.

El uso de los dispositivos móviles nos ha permitido realizar tareas en lugares que antes no eran los adecuados para ellas debido a varios factores, la realidad virtual era una herramienta a la que solo se podía acceder desde un computador de mesa con características de alto rendimiento, pero

este panorama ha cambiado, desde un Smartphone se puede acceder a una experiencia de realidad virtual de entrada pero así mismo estos dispositivos también deben contar con especificaciones aptas para su uso como lo es la tasa de refresco de la pantalla o la capacidad de RAM con la que cuenta el dispositivo. El avance tecnológico nos ha permitido gozar de estas herramientas sin pensar en factores que antes se convertían en excusas y por eso a día de hoy las restricciones se pueden percibir como algo con menor importancia a como lo eran antes dejando a casi cualquier individuo tener una experiencia decente.

Los videojuegos han crecido como un medio de entretenimiento aclamado por muchos y esto también trajo nuevas tecnologías a la industria como lo es la plataforma de desarrollo Unity que no solo permite crear juegos también es apta para la construcción de aplicaciones y más en RV o RA trayendo al mercado más productos y avances tecnológicos, pero claro está que para poder acceder a ellos se debe contar con un dispositivo de gama media tanto en computadores como en dispositivos inteligentes.

La tecnología avanza y la potencia bruta de un dispositivo cada vez es mejor pero así mismo el software se vuelve más demandante en cálculos, por eso se debe contar con un instrumento que esté a la par de su demanda.

#### **4.5. Suposiciones y dependencias**

El software solo puede ser instalado y ejecutado en sistemas operativos Windows 10 o Android 7.0 en adelante esto es debido a que Android soporta la función de Realidad Virtual es a partir de la versión Nougat.

El Framework para funcionar correctamente y evitar lentitud o crasheo (la aplicación deja de funcionar de forma inesperada) debe ser ejecutado en dispositivos que cuenten con 4 GB de memoria RAM o más, esto es debido a la alta carga de objetos en simultáneo junto con la función de RV, lo que llega a ocupar significativamente espacio en la memoria.

No es necesario una conexión a internet para ejecutar la aplicación gracias a que no se consumen servicios o protocolos para intercambiar información, la información se administra directamente en el almacenamiento interno ya sea del Smartphone o computador.

Para hacer uso de la función importar objeto 3D es necesario acatar los formatos permitidos por la aplicación preferiblemente utilizar el renderizado en OBJ para evitar inconvenientes en la carga y escala del objeto en cuestión esto permitirá al usuario tener una agradable experiencia de uso.

## 4.6. Requisitos específicos

### 4.6.1. Actores/roles.

Tabla 1.

*Actor del software*

Actor / nombre:	Usuario
Descripción:	El actor representa a un usuario primario que necesita crear un tour de realidad virtual.
Características:	El usuario debe tener conocimientos básicos en el uso de aplicaciones móviles, de escritorio y en modelos 3D. Este usuario se encuentra en un rango de edad de 18 a 50 años.
Relaciones:	Operador del Framework
Referencias:	En el caso de uso Crea, edita, reproduce
Autor:	Leonardo Vélez García

*Nota:* (autoría propia).

Tabla 2.

*Atributos del actor, su descripción y el tipo del atributo*

Atributos		
Nombre	Descripción	Tipo
Edad	18 años a 50 años para usar el modo editor y desde los 12 años para reproducir las escenas.	Primario
Género	Sin restricción.	Secundario
Conocimiento Tecnológico	Uso promedio de una aplicación móvil. (Redes sociales, editor de fotos etc.). Conocimiento en renderizado de objetos en formato OBJ.	Primario

Educación	Superior	Primario
-----------	----------	----------

*Nota:* (autoría propia).

#### 4.6.2. Requisitos funcionales.

Tabla 3.

*Requisitos funcionales*

Requisito funcional No.	Requisito
RF 1	La aplicación permitirá que el usuario pueda importar diseños de renderizado en 3D o fotos 360°.
RF 2	La aplicación contará con diseños predeterminados de objetos 3D.
RF 3	El usuario podrá elegir un diseño predeterminado.
RF 4	El usuario podrá crear sus escenas.
RF 5	La aplicación permitirá editar las escenas hechas por el usuario.
RF 6	El usuario podrá reproducir un diseño de escena o tour hecho por él.

*Nota:* (autoría propia).

4.6.3. Diagrama de casos de uso.

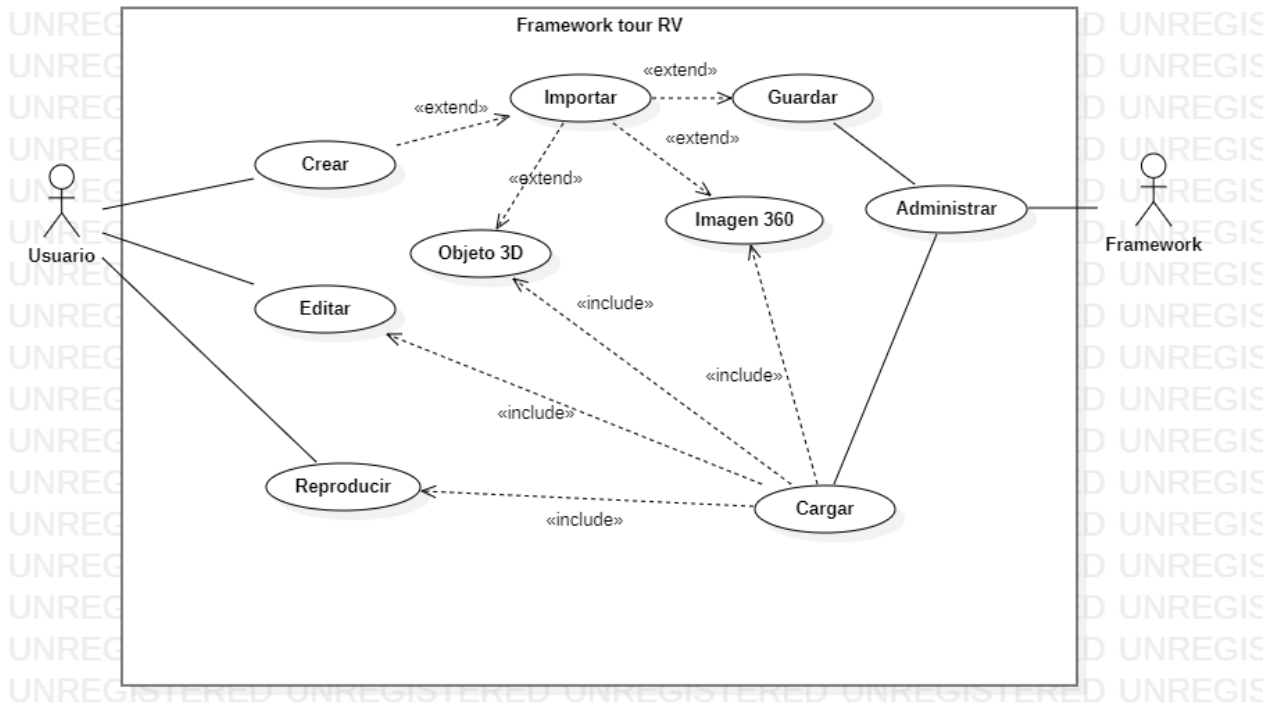


Figura 2. Diagrama de caso de uso. (Autoría propia).

4.6.4. Especificaciones de los casos de uso.

Tabla 4.

Primer caso de uso

Nombre del caso de uso:	Crear
Actores:	Usuario y Framework
Tipo de caso de uso:	Primario
Referencias:	RF4
Precondición:	Iniciar la aplicación
Poscondición:	Crea escenas en blanco
Autor:	Leonardo Vélez García

Nota: (autoría propia).

Tabla 5.

*Propósito del primer caso de uso*

Propósito:
El usuario crea un tour en blanco para su futura edición

*Nota:* (autoría propia).

Tabla 6.

*Resumen del primer caso de uso*

Resumen:
El usuario necesita o desea diseñar y crear un Tour de realidad virtual, el Framework crea el escenario para que el usuario pueda editar a su necesidad o gusto las escenas.

*Nota:* (autoría propia).

Tabla 7.

*Flujo principal del primer caso de uso*

Flujo principal	
1	El usuario oprime botón crear.
2	La aplicación pide digitar nombre y datos del tour.
3	El usuario digita la información solicitada.
4	El usuario oprime el botón de guardar.
5	La aplicación muestra al usuario la interfaz de edición

*Nota:* (autoría propia).

Tabla 8.

*Flujo alterno del primer caso de uso*

Flujo alterno	
1a	La aplicación muestra cuadros para insertar texto.



2a	El usuario digita nombre y datos del tour.
2b	El usuario oprime el botón cancelar.
3a	La aplicación almacena temporalmente la información digitada.
4a	La aplicación carga las escenas

*Nota:* (autoría propia).

Tabla 9.

*Segundo caso de uso*

Nombre del caso de uso:	Editar
Actores:	Usuario y Framework
Tipo de caso de uso:	Primario
Referencias:	RF1, RF2, RF3, RF5
Precondición:	Crear o cargar un tour
Postcondición:	Guardar o reproducir
Autor:	Leonardo Vélez García

*Nota:* (autoría propia).

Tabla 10.

*Propósito del segundo caso de uso*

Propósito:
El sistema permitirá al usuario editar escenas para la creación de un tour

*Nota:* (autoría propia).

Tabla 11.

*Resumen del segundo caso de uso*

Resumen:
----------

El usuario elige la cantidad de escenas que conformará el tour junto con los objetos que estarán en las escenas y las respectivas imágenes, pondrá los puntos de control a su gusto para conectar y conformar el recorrido.

Nota: (autoría propia).

Tabla 12.

*Flujo principal del segundo caso de uso*

Flujo principal	
1	El usuario agrega escenas.
2	La aplicación muestra las escenas agregadas.
3	El usuario agrega puntos de control.
4	La aplicación muestra los puntos de control.
5	El usuario oprime el botón agregar objeto 3D
6	La aplicación muestra un panel donde están los objetos que se pueden agregar o importar.
7	El usuario oprime botón guardar
8	La aplicación muestra panel de guardado
9	El usuario oprime botón reproducir
10	La aplicación reproduce las escenas.
11	El usuario oprime el botón salir.
12	La aplicación abre el menú de crear o cargar.

Nota: (autoría propia).

Tabla 13.

*Flujo alternativo del segundo caso de uso*

Flujo alternativo	
1a	El usuario decide cuántas desea agregar.
2a	El usuario oprime el botón para moverse entre escenas.
3a	El usuario decide cuántos puntos de control colocar y su ubicación.
5a	La aplicación carga el panel de objetos e importación.
6a	El usuario elige un objeto.
8a	El usuario elige slot para guardar.
8b	El usuario cancela.
9a	El sistema carga las escenas en modo reproducción.
10a	El usuario sale de modo reproducir.
11a	La aplicación pregunta si desea continuar con la acción de salir.

Nota: (autoría propia).

Tabla 14.

*Tercer caso de uso*

Nombre del caso de uso:	Reproducir
Actores:	Usuario y Framework
Tipo de caso de uso:	Primario
Referencias:	RF6
Precondición:	Cargar Tour
Postcondición:	Abrir una escena

Autor:	Leonardo Vélez García
--------	-----------------------

Nota: *(autoría propia)*.

Tabla 15.

*Propósito del tercer caso de uso*

Propósito:
Permitir al usuario reproducir en realidad virtual el tour

Nota: *(autoría propia)*.

Tabla 16.

*Resumen del tercer caso de uso*

Resumen:
Cuando el usuario tenga un Tour listo para exhibir o para observar el progreso sólo entrará al modo Ver en RV, el sistema cargará las escenas y el usuario podrá empezar el recorrido.

Nota: *(autoría propia)*.

Tabla 17.

*Flujo principal del tercer caso de uso*

Flujo principal	
1	El usuario oprime el botón reproducir para pasar al modo RV.
2	La aplicación aconseja poner el celular en un visor de RV.
3	El usuario interactúa con los puntos de control para moverse entre escenas.
4	La aplicación responde a los movimientos del usuario.

Nota: *(autoría propia)*.

Tabla 18.

*Flujo alternativo del tercer caso de uso*

Flujo alternativo
-------------------

1a	La aplicación carga las escenas en modo realidad virtual.
2a	El usuario pone su dispositivo en el visor.
2b	El usuario conecta el visor al computador.
3a	La aplicación carga las siguientes escenas
4a	El usuario mueve la cabeza para observar su entorno
4b	El usuario oprime el botón salir del modo RV.

Nota: (autoría propia).

#### 4.7. Requisitos de rendimiento

Para crear un Tour solo se necesita un usuario, esa persona puede importar objetos 3D o imágenes para hacer la conformación de las escenas, así mismo cuando desee podrá guardar o cargar lo realizado si necesita de ello, cabe aclarar que todas las funciones se hacen una a la vez, ninguna de ellas se puede realizar en simultáneo debido a la carga que conlleva realizar las tareas.

Los procesos de carga y guardado pueden tardar máximo 5 segundos dependiendo del rendimiento del dispositivo, las tareas de poner punto de control o crear escena se emplean en un lapso de 1.5 segundos dependiendo de la carga que haya en el momento.

Para importar objetos en 3D el tiempo de búsqueda y carga del objeto varía dependiendo del peso o carga poligonal que tenga el mismo, aproximadamente 10 segundos para 750.000 triángulos que conforman el objeto, para un elemento de menor composición poligonal la carga es inmediata.

Para cargar o guardar los progresos de la creación de tours la aplicación solo cuenta con cuatro slots (espacios) de almacenamiento debido a que si el usuario construye un recorrido con demasiados objetos hará que el Tour pese significativamente y así misma tarde en cargar.

#### 4.8. Restricciones de diseño

El programa al ser un complemento de otro software debía tener afinidad en los menús por ello se utilizó el mismo diseño que se implementó en la App VRI-Agustiniana de esta manera la aplicación en diseño y funcionalidad será intuitiva mostrando al usuario familiaridad que existe entre las aplicaciones, en el mejor de los casos no se notara que se trata de dos aplicaciones que funcionan en equipo.

Para los botones que conforman la interfaz de usuario en principio se utilizaron iconos negros

con fondo transparente para dar una sensación de minimalismo y elegancia, pero debido al uso de fotografías en 360° los botones se perdían de vista por los contrastes en lugares oscuros o de tonalidad negra que se notaban al observar las imágenes, para solucionar este inconveniente el fondo transparente de los iconos se rellenó de color blanco.

#### 4.9. Atributos del software del sistema

Este software no tiene restricciones de usuario, una de las principales razones de esto es que cualquier usuario puede realizar las tareas que permite el software, el único mecanismo de seguridad y privacidad con el que cuenta el Framework es en la búsqueda de objetos dentro del almacenamiento interno, para evitar mostrar información o documentos no deseados la aplicación tiene la opción elegir el tipo de archivo que se desea importar y así enseña solo los archivos con el formato permitido para cargar, de esta manera se evita exhibir archivos privados también facilitando la búsqueda del documento.

Este programa tiene una alta portabilidad debido a que se puede ejecutar en Smartphone con sistema operativo Android o portátiles con Windows, aun así si el usuario desea puede ejecutar en un computador de escritorio.

El mantenimiento es casi nulo gracias a que no se necesita de servidores para almacenar o cargar información, simplemente se debe contar con el espacio mínimo requerido para tener instalada la aplicación.

En cuanto a la fiabilidad si se presenta un fallo, error o una carga de objetos mal realizada lo ideal es evitar guardar el progreso realizado y cargar desde el anterior guardado esto permitirá al Framework traer con normalidad las escenas.

Tabla 19.

#### *Requisitos no funcionales*

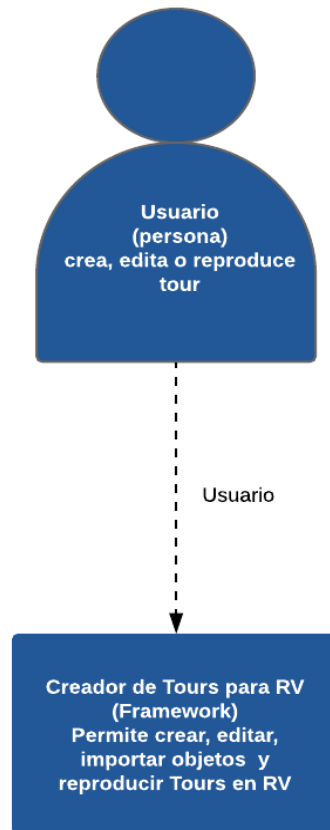
Requisito no funcional No.	Requisito
RNF 1	La aplicación será ejecutable para dispositivos Android y Windows.
RNF 2	La aplicación se podrá ejecutar desde dispositivos Android desde la versión 7.0 o superior.

RNF 3	La aplicación podrá ejecutarse en dispositivos con 4GB de RAM o superior.
RNF 4	La aplicación podrá ejecutarse sin conexión a internet.
RNF 5	La aplicación solo permite importar renderizado de formato OBJ o FBX
RNF 6	La aplicación solo permite importar fotos de formato JPG o PNG

Nota: *(autoría propia)*.

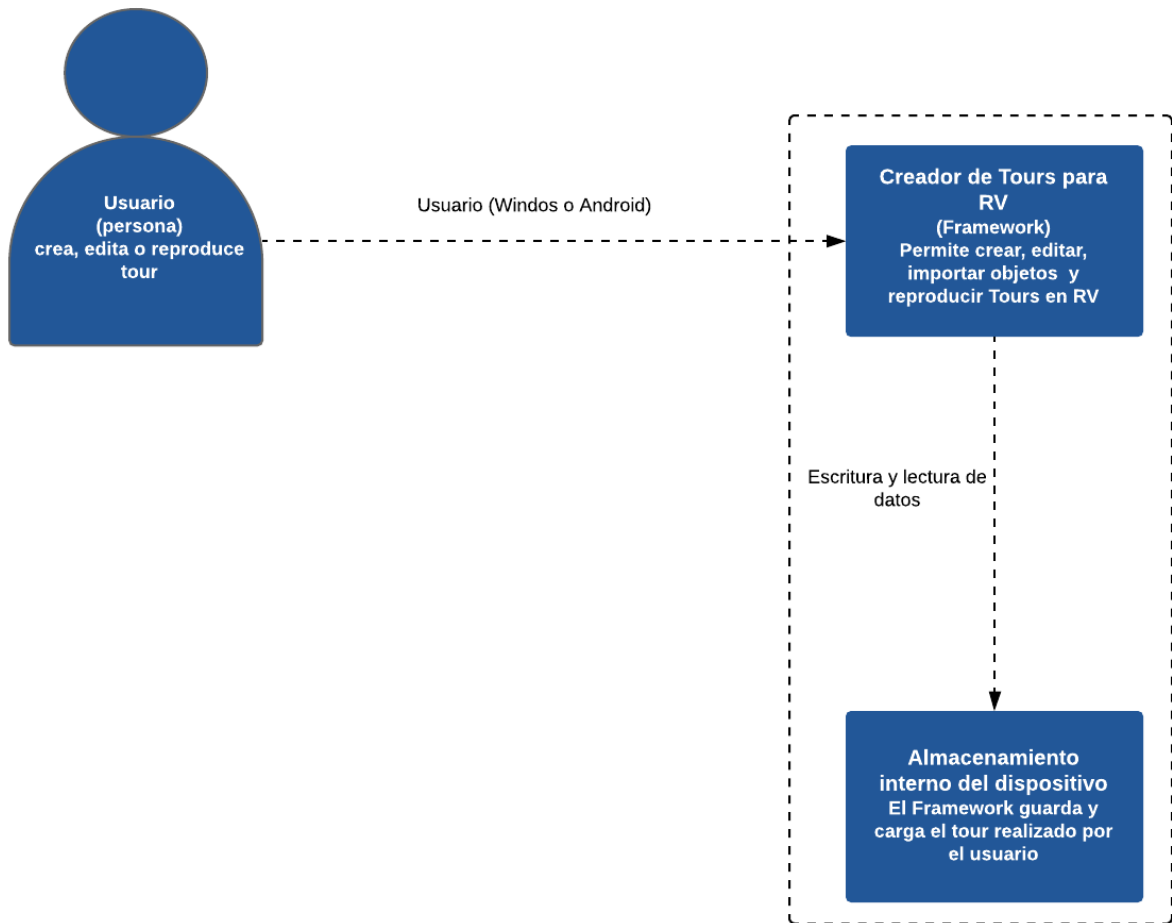
## 5. Diseño del software (ISO -12207-1)

### 5.1. Diseño de la arquitectura de software

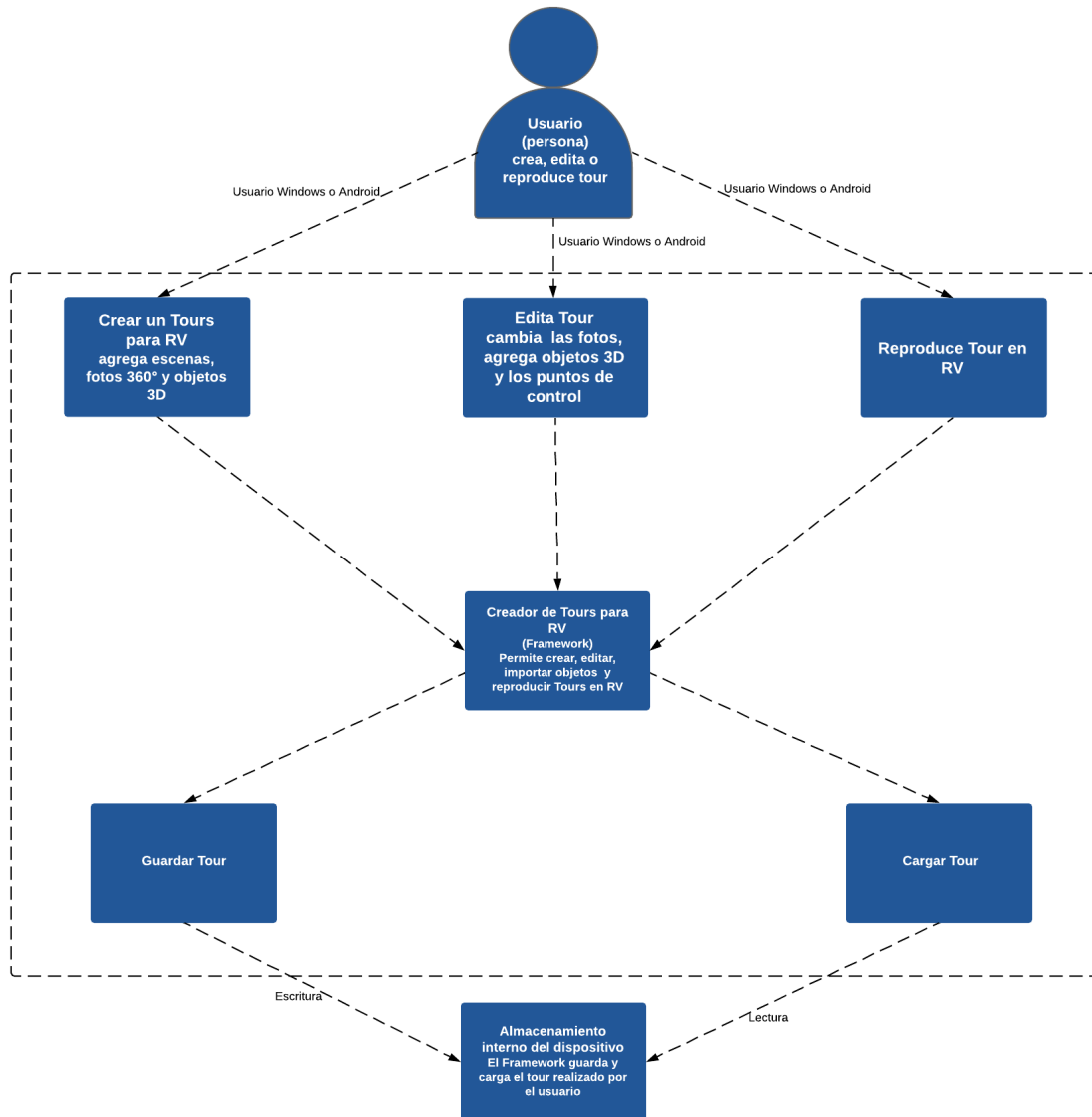


**Figura 3.** Arquitectura de software modelo C4 primer nivel. Autoría propia.





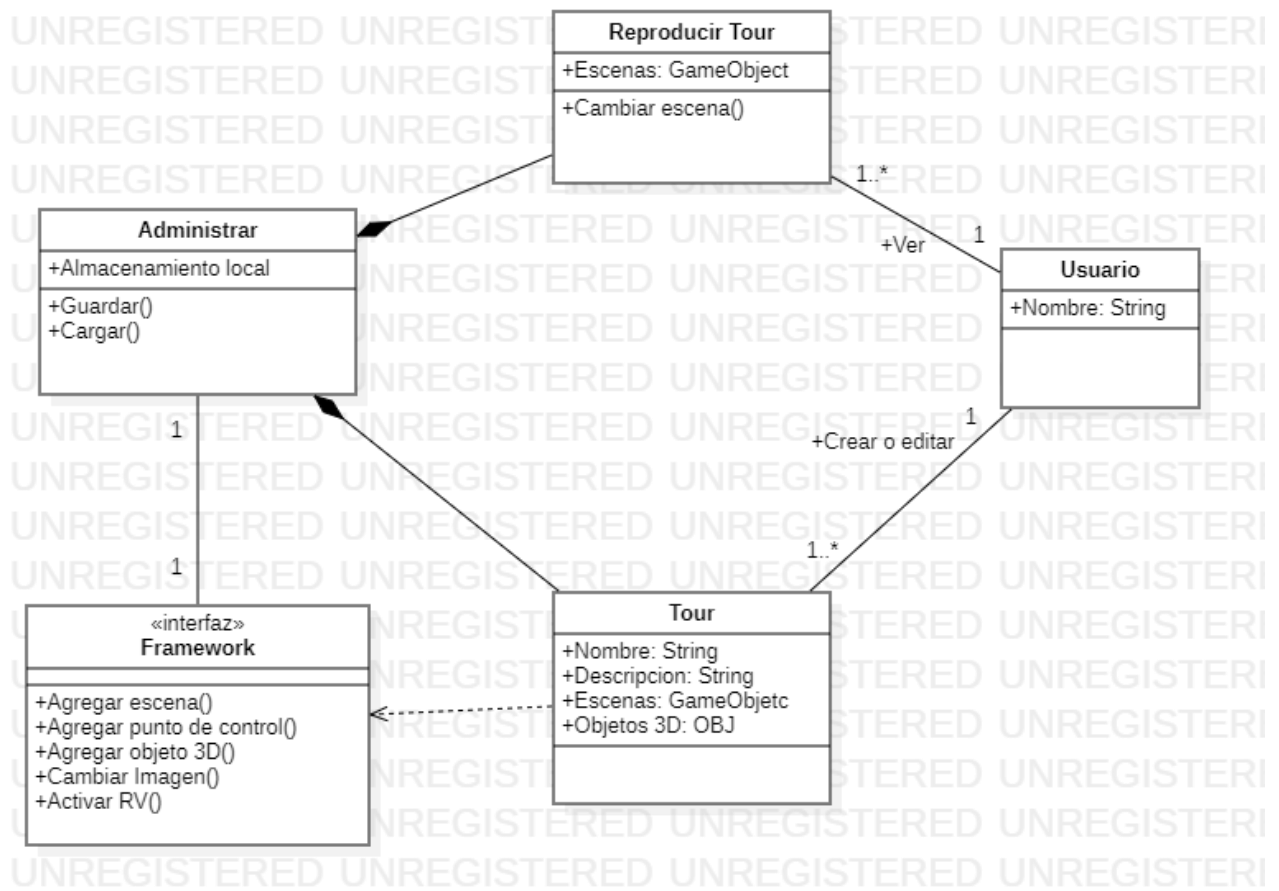
**Figura 4.** Arquitectura de software modelo C4 segundo nivel. Autoría propia.



**Figura 5.** Arquitectura de software modelo C4 tercer nivel. Autoría propia.

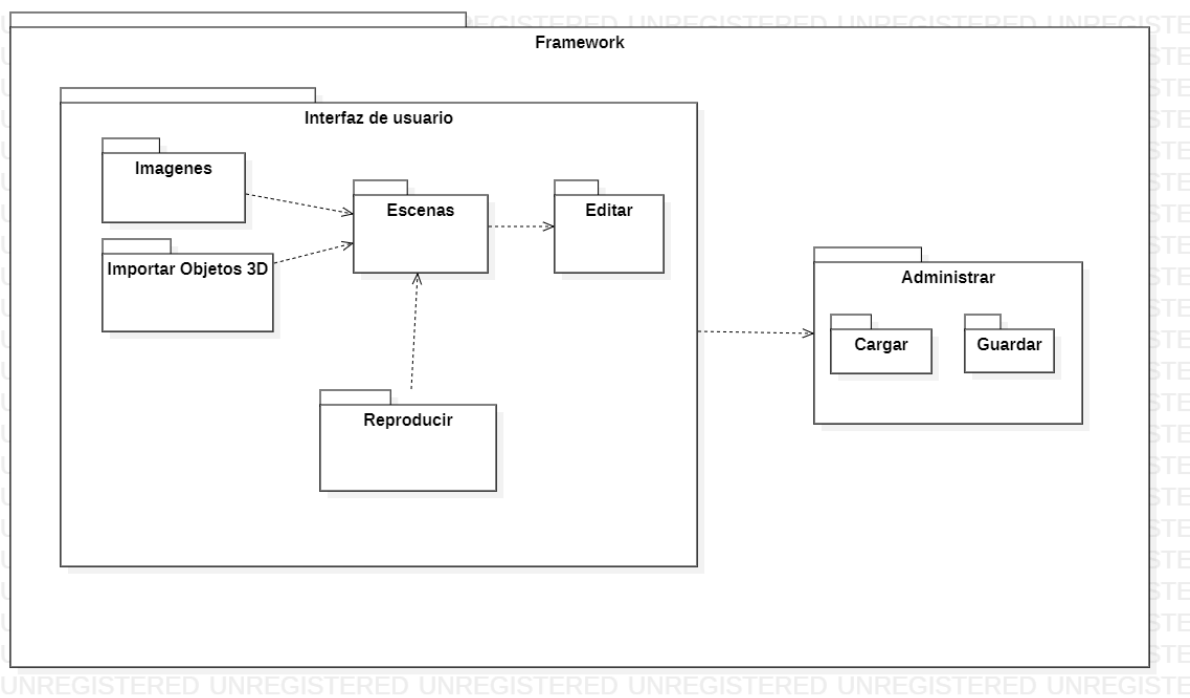
## 5.2. Diseño detallado del software

### 5.2.1. Diagrama de clases.



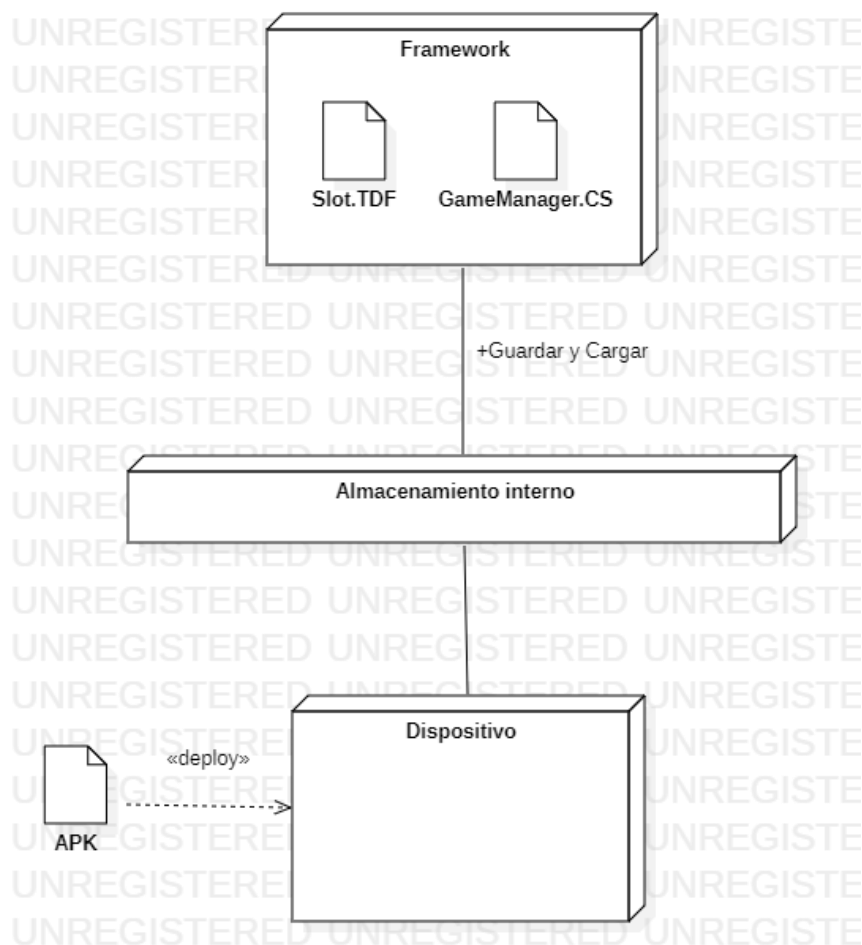
**Figura 6.** Diagrama de clases. Autoría propia.

### 5.2.2. Diagrama de paquetes.



**Figura 7.** Diagrama de paquetes. Autoría propia.

### 5.2.3. Diagramas de despliegue.



**Figura 8.** Diagrama de despliegue. Autoría propia.

### 5.3. Diseño de la interfaz

#### 5.3.1. Interfaz gráfica de usuario.



**Figura 9.** Interfaz gráfica del Framework. Autoría propia.

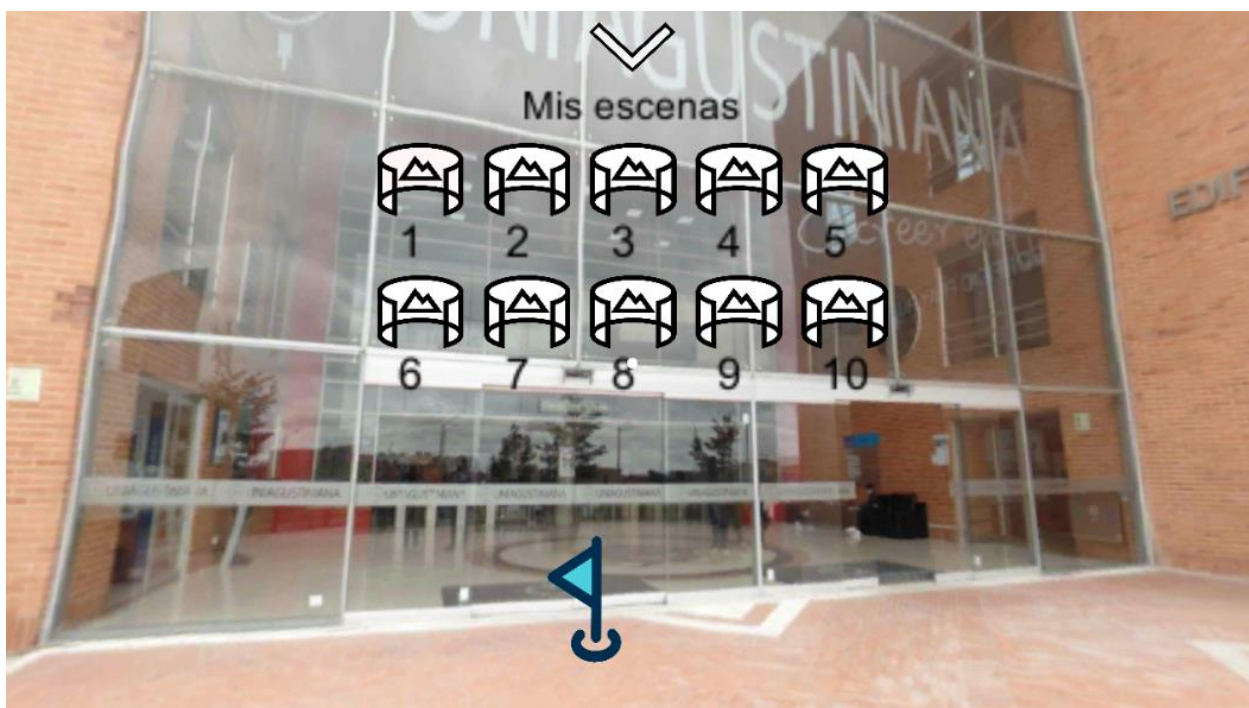
El usuario cuenta con la siguiente estructura de interfaz:

Botón 1 llamado “Agregar escena” su función es aumentar las escenas editables para el usuario (si llega al máximo saltara un texto de aviso).



**Figura 10.** Interfaz gráfica botón 1 “Agregar escena”. Autoría propia.

Botón 2 llamado “Mis escenas” este botón despliega la lista de escenas agregadas por el usuario para que él pueda desplazarse entre ellas.



**Figura 11.** Interfaz gráfica botón 2 “Mis escenas”. Autoría propia.

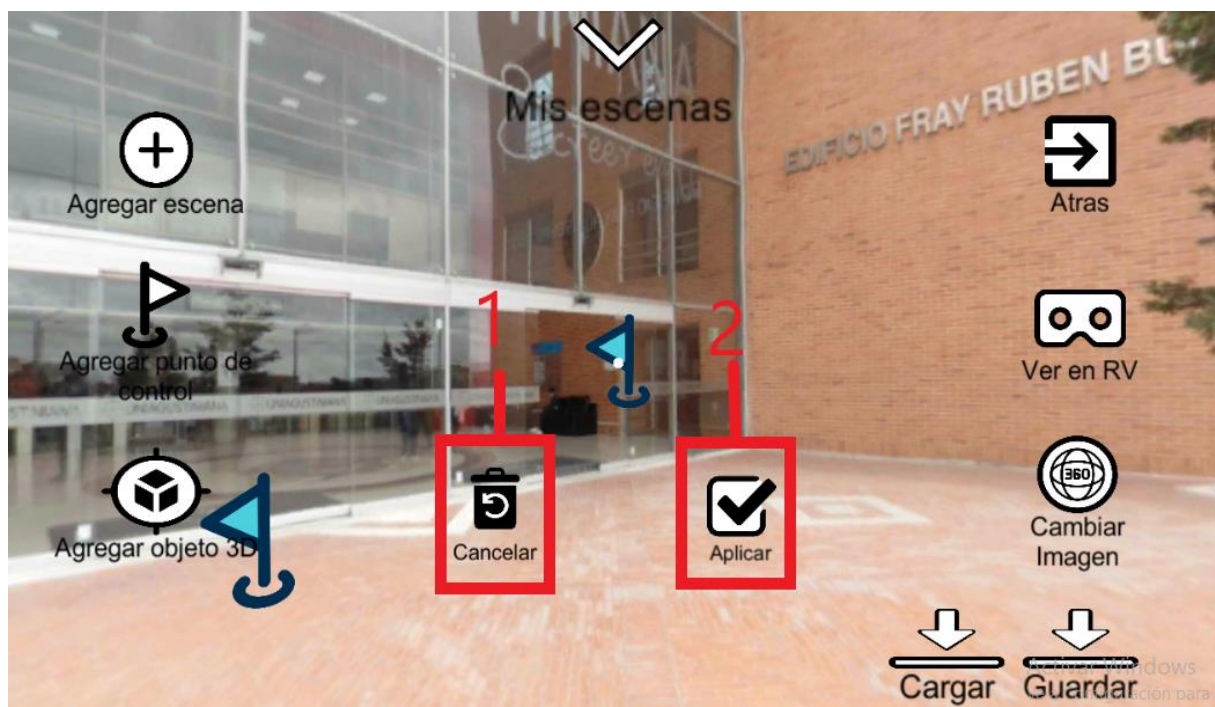
Botón 3 llamado “Agregar punto de control”, al dar clic en él se abre un panel que muestra los



botones para elegir qué punto de control desea poner en la escena.



**Figura 12.** Interfaz gráfica botón 3 “Panel Agregar punto de control”. Autoría propia.

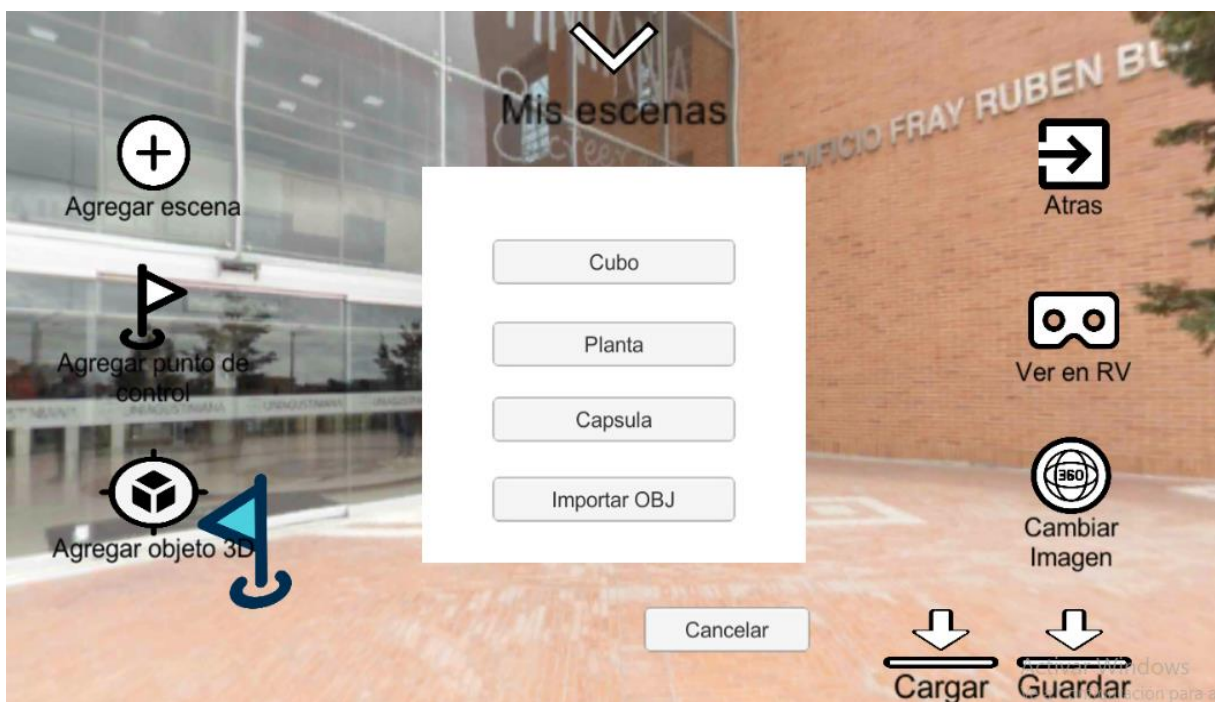


**Figura 13.** Interfaz gráfica botón 3 “Agregar punto de control”. Autoría propia.

Con el botón cancelar se cancela la acción de colocar el punto de control y con el de aplicar se sitúa el punto de control donde la retícula (punto blanco) esté apuntando.



Botón 4 llamado “Agregar objeto 3D”, cuando se interactúa con el abre un panel que permite poner un objeto predeterminado o exportar uno del almacenamiento interno del dispositivo.



**Figura 14.** Interfaz gráfica botón 4 “Panel Agregar objeto 3D”. Autoría propia.

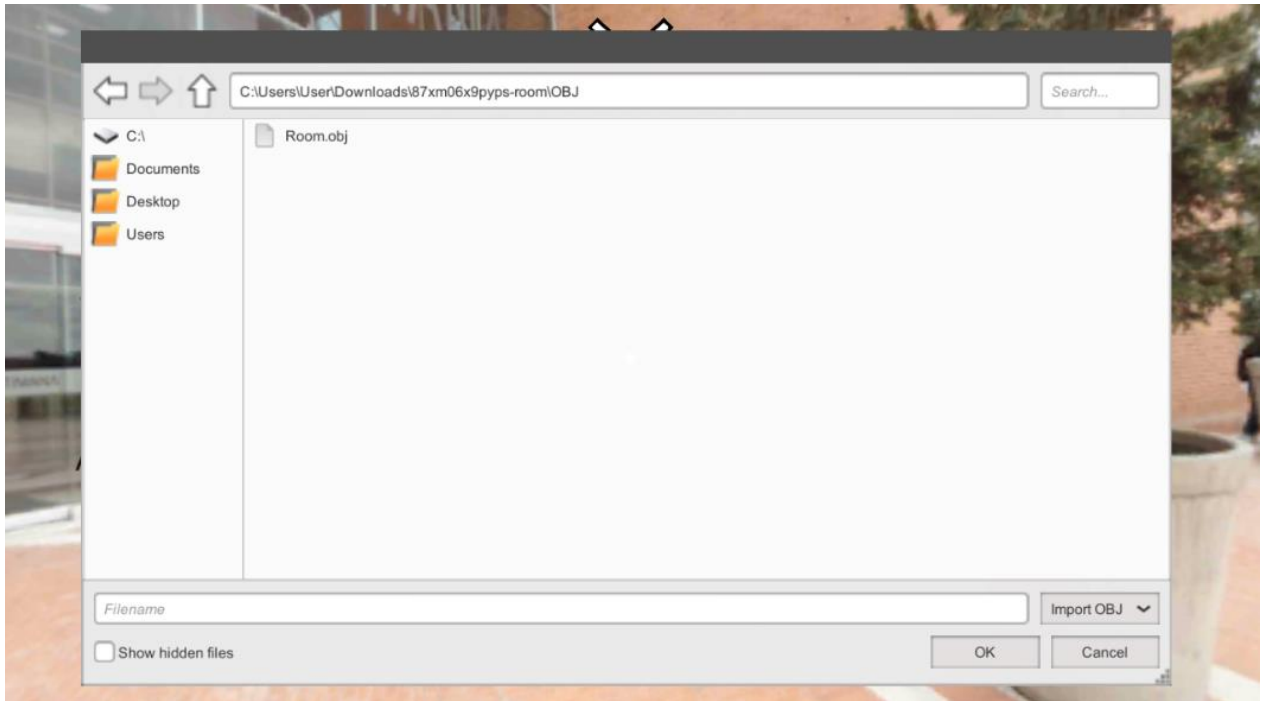
Al elegir un objeto predeterminado se mostrará en pantalla de la siguiente manera:



**Figura 15.** Interfaz gráfica botón 4 “Agregar objeto 3D”. Autoría propia.

Para colocar el objeto solo bastará con realizar la misma acción que con el punto de control o si desea cancelar se detendrá la tarea.

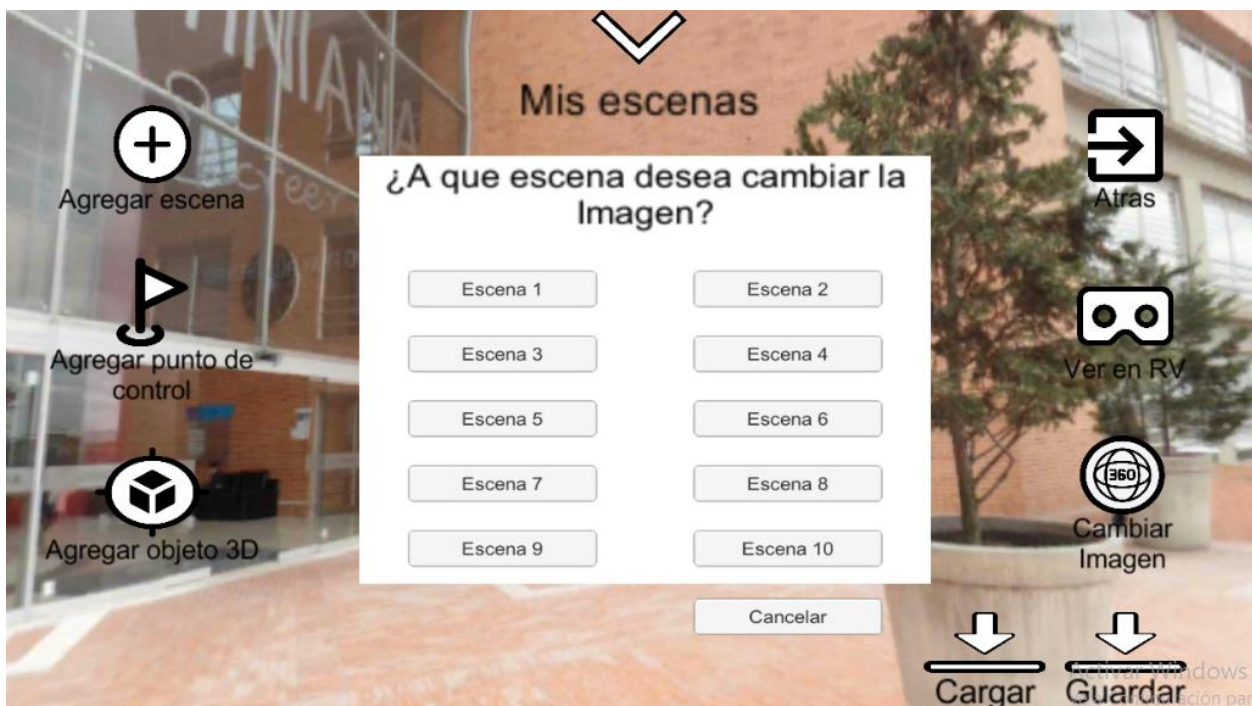
Si el usuario eligió la opción de importar un objeto 3D se mostrará el siguiente explorador de archivos:



**Figura 16.** Interfaz gráfica botón 4 “Agregar objeto 3D con explorador de archivos”. Autoría propia.

Para ver el objeto 3D en la escena solo deberá buscarlo en el explorador y seleccionarlo dando clic en el botón “ok”.

Botón 5 llamado “Cambiar imagen”, abre un panel donde se podrá elegir qué imagen predeterminada poner en la escena o también permite explotar una foto almacenada en el dispositivo.



**Figura 17.** Interfaz gráfica botón 5 “Panel 1 Cambiar imagen”. Autoría propia.

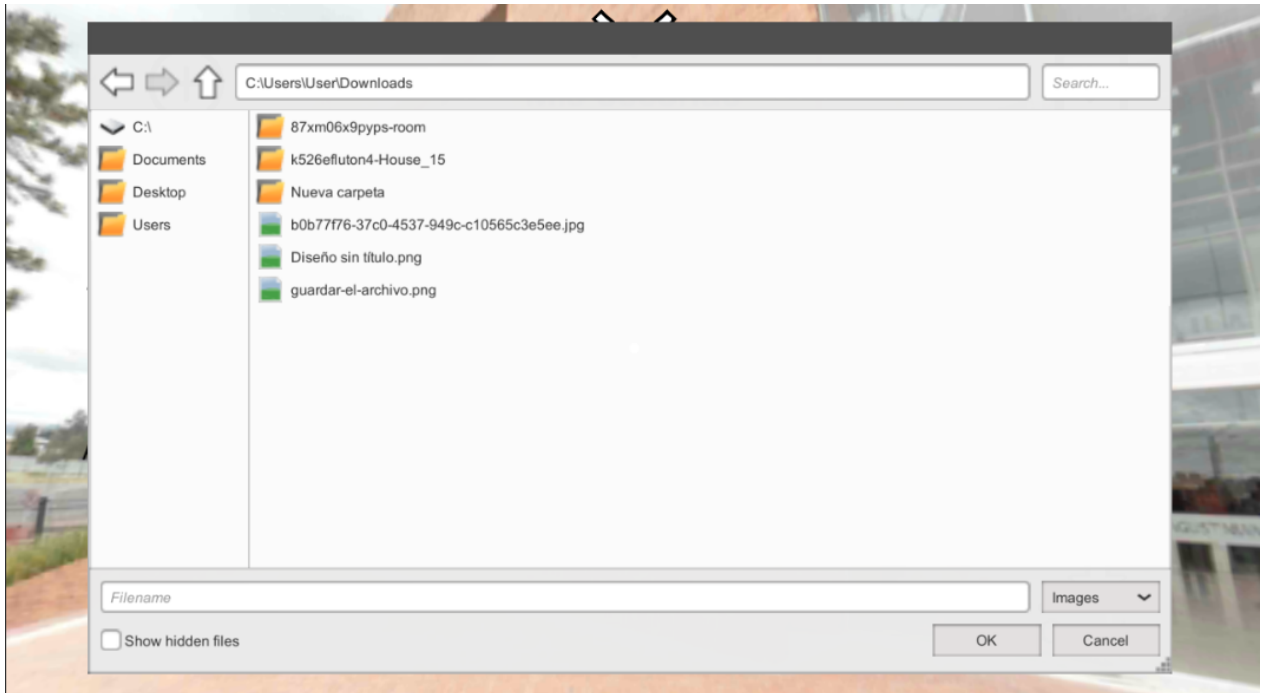
Luego de elegir la escena a la que se desea cambiar la imagen 360° se mostrará el siguiente panel donde se cuenta con imágenes predeterminadas o la opción de importar una



**Figura 18.** Interfaz gráfica botón 5 “Panel 2 Cambiar imagen”. Autoría propia.

Si el usuario elige importar se desplegará el siguiente explorador de archivos:





**Figura 19.** Interfaz gráfica botón 5 “Cambiar imagen con explorador de archivos”. Autoría propia.

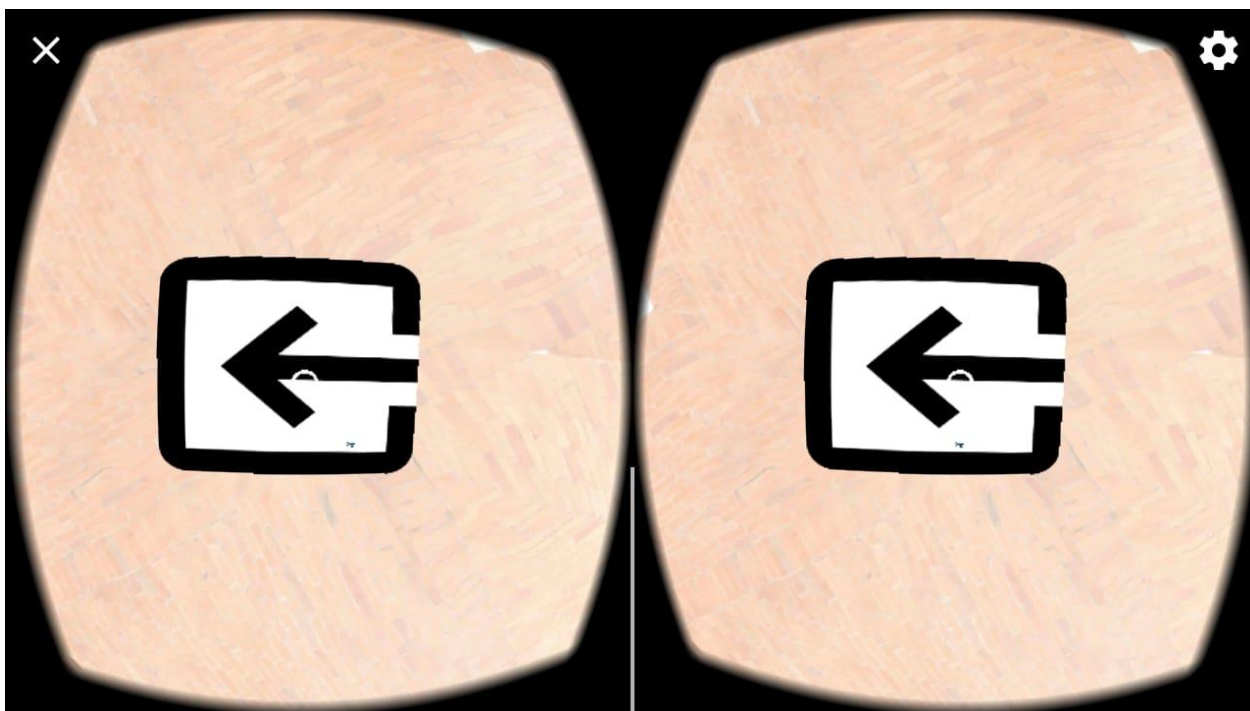
En este explorador podrá buscar la imagen que desea aplicar (el explorador funciona de igual manera en los dos tipos de dispositivos permitidos para correr la aplicación).

Botón 6 llamado “Ver en RV”, activa el reproductor de realidad virtual donde se podrá realizar el recorrido (Para disfrutar adecuadamente del recorrido creado es ideal utilizar un visor)



**Figura 20.** Interfaz gráfica botón 6 “Ver en RV”. Autoría propia.

Para salir del modo Ver en RV el usuario debe mirar hacia el piso, allí encontrará el icono de atrás, al interactuar volverá a el modo editor.



**Figura 21.** Interfaz gráfica botón 6 “Salir del modo Ver en RV”. Autoría propia.

Botón 7 llamado “Guardar” aquí el usuario podrá almacenar los avances hechos en cada escena, se despliega un panel donde puede elegir en qué ranura de guardado depositara su recorrido.



**Figura 22.** Interfaz gráfica botón 7 “Panel Guardar”. Autoría propia.

Al elegir cualquiera de los siguientes Slot de almacenamiento se procederá a almacenar el Tour.

Para cargar un tour simplemente se debe oprimir el botón cargar que desplegará el mismo panel de guardar, pero en este caso su función será cargar el tour.

Al elegir cualquiera de los siguientes Slot de almacenamiento se procederá a almacenar el Tour.

Botón 8 llamado “Atrás” este botón permite al usuario volver al menú principal.



**Figura 23.** Interfaz gráfica botón 8 “Atrás”. Autoría propia.

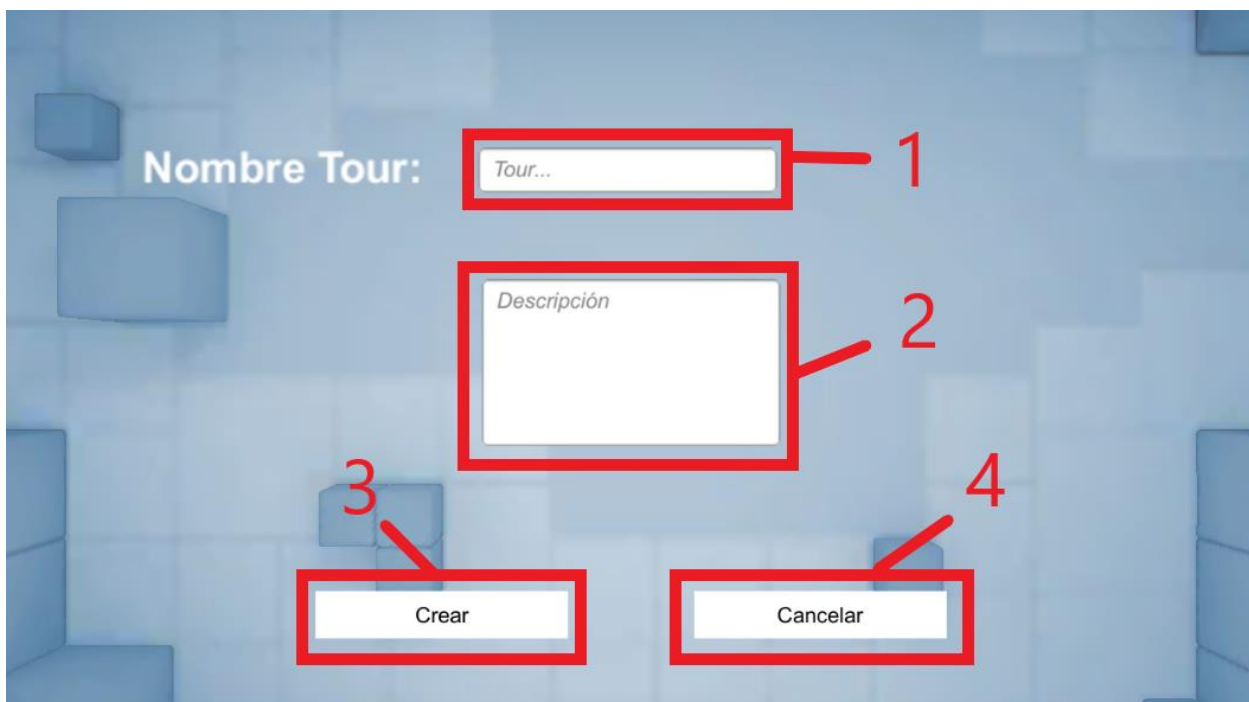
Mostrará esta pantalla antes de continuar, si selecciona “No” seguirá en el modo editor y si selecciona la opción “Si” lo llevara al menú principal.



**Figura 24.** Interfaz gráfica botón 8 “Atrás” menú principal. Autoría propia.



### Interfaces de entrada.



**Figura 25.** Interfaz de entrada. Autoría propia.

El Framework cuenta con esta distribución en la interfaz de entrada.

En el cuadro 1 se debe digitar el nombre que tendrá el Tour.

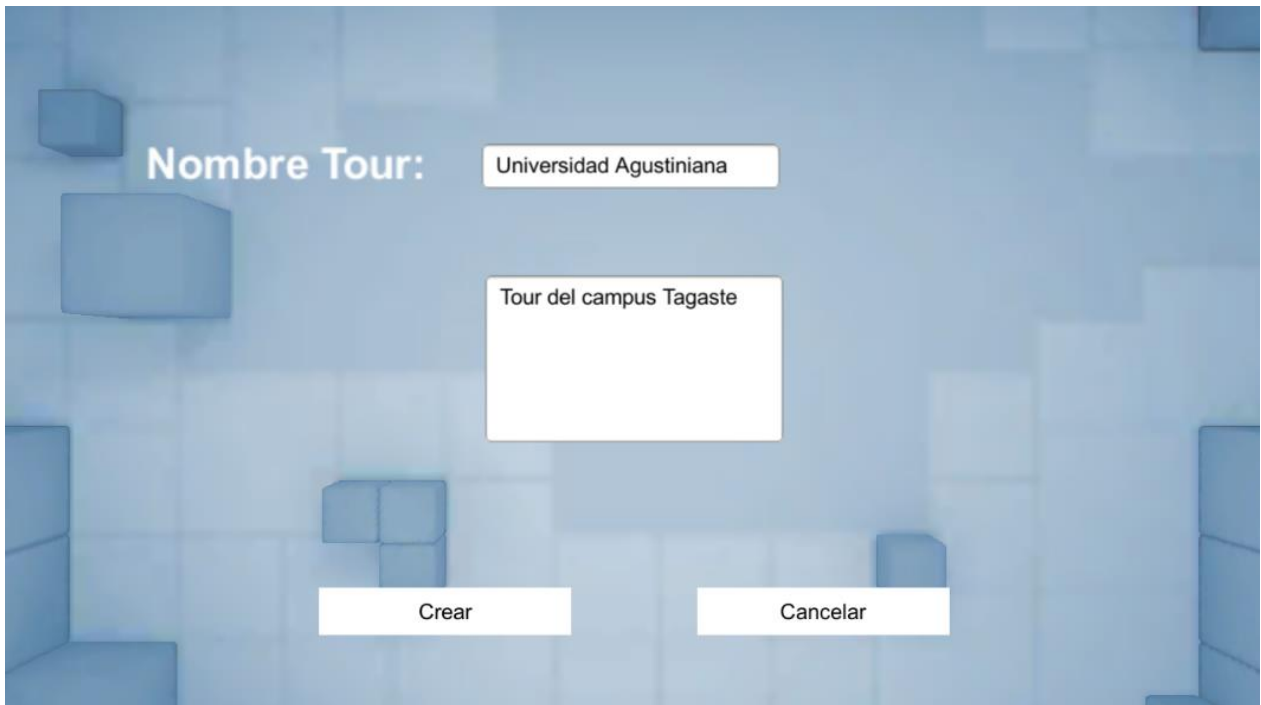
En el cuadro 2 el usuario deberá digitar una descripción del recorrido (este dato es opcional).

En el cuadro 3 se encuentra el botón “Crear” al oprimir se cargará el editor con la escena 0 y su respectiva interfaz, el usuario solo podrá dar clic cuando haya ingresado el nombre del tour.

En el cuadro 4 está el botón “Cancelar” que permite al usuario regresar al menú principal.

Así se verían los cuadros de texto con su respectiva información.





**Figura 26.** Interfaz de entrada ejemplo. Autoría propia.

### **Interfaces de salida.**

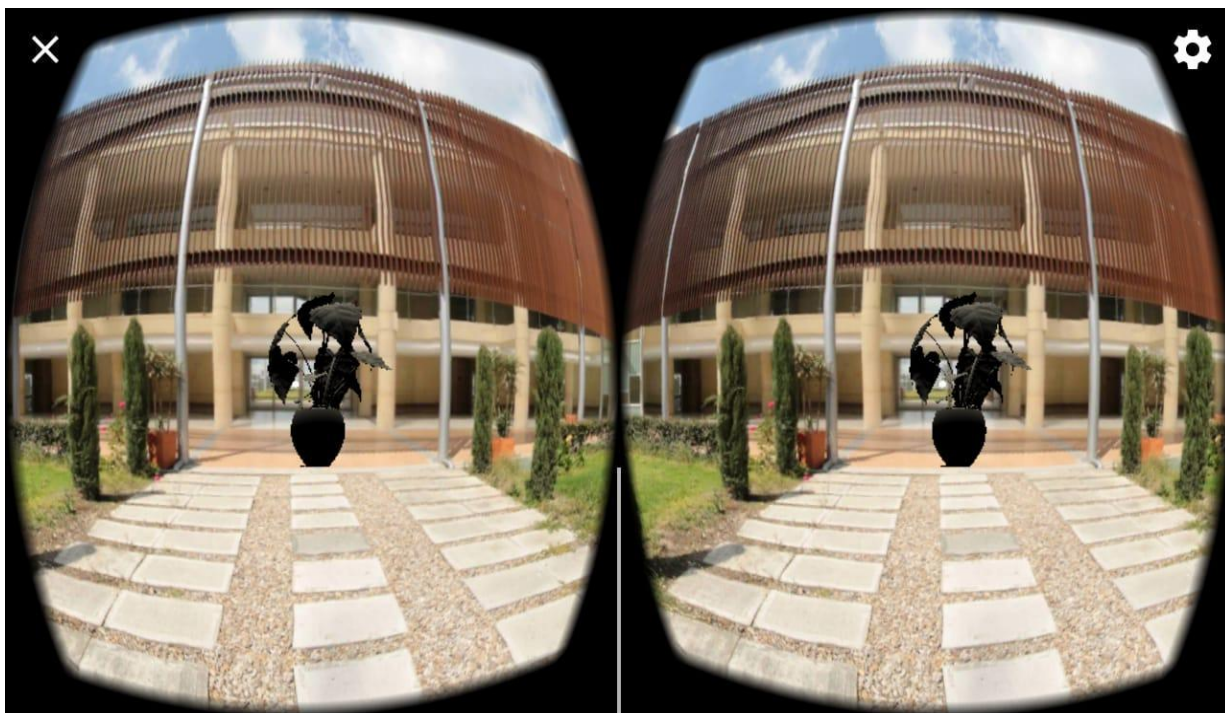
La interfaz de salida cuenta con dos tipos, la primera es la percepción del usuario de cómo va quedando la escena a medida que agrega objetos o cambia de imagen desde el modo editor, la segunda foto de interfaces de salida es el acabado final de la misma en el modo de Realidad Virtual y la última imagen es la interfaz de salida en modo “ver en RV” desde un Smartphone.



**Figura 27.** Interfaz de salida modo editor. Autoría propia.



**Figura 28.** Interfaz de salida Realidad Virtual PC. Autoría propia.



**Figura 29.** Interfaz de salida Realidad Virtual Smartphone. Autoría propia.

## **6. Implementación**

### **6.1. Plataformas de desarrollo**

Como plataformas de desarrollo se utilizó el motor gráfico Unity para crear el Framework, se eligió este entorno ya que permite utilizar la tecnología de Realidad Virtual y así mismo alternar a un ambiente 3D donde el usuario pueda editar con comodidad el recorrido para después reproducir activando el RV.

Unity se ejecutó en una computadora con Windows 10, el motor gráfico da la posibilidad de crear juegos o aplicaciones para diversos sistemas por lo cual es ideal para la creación del editor permitiendo correr el mismo en los sistemas operativos Android y Windows 10.

Como esta aplicación no necesita de conexión a internet ni almacenar mucha información o agrupar la misma para su fácil acceso no se necesitó del uso de una base de datos ni de una conexión cliente servidor para almacenar o solicitar dicha información, el programa se encarga de crear su respectivo archivo para guardar y cargar lo realizado por el usuario.

El Framework al ser un agregado del prototipo App VRI-Agustiniana se tenía que desarrollar en la misma plataforma para hacer su total compatibilidad y así permitirle al usuario tener la función opcional de crear un recorrido de Realidad Virtual inmersiva.

Para programar el software se utilizó el lenguaje C# debido a que es el más apto y optimizado para desarrollar en el motor gráfico Unity, el código se escribía y se compilaba en Visual Studio conformando así un entorno ideal para la creación de la aplicación.

Los modelos de objetos en 3D utilizados en las pruebas del Framework se sacaron como recursos de internet concretamente de la página free3D.com donde se consiguieron modelos tanto de gran tamaño como pequeños en renderizado por polígonos, ese recurso fue de gran ayuda para determinar los parámetros correctos con los que debe contar el software y así ofrecer una mejor experiencia de usuario.

Para las fotos en 360° se utilizaron los recursos ofrecidos por la universidad implementados en proyectos anteriores, las fotos se implementaron como texturas de las escenas y así complementar mejor con los objetos 3D.

### **6.2 Base de datos**

Este prototipo no cuenta con base de datos por lo tanto la escritura y lectura de la información se realizan directamente en el almacenamiento interno del dispositivo, en una carpeta nombrada DataSaveTour se encuentran los archivos de tipo “.txt” (este tipo de archivo es ideal para el manejo

de datos del entorno creado en Unity) en concreto son 4 archivos cada uno se encarga de conservar lo entregado en los Slots (ranuras de almacenamiento). Un diccionario se encarga de almacenar los datos de las escenas y cuenta con la siguiente estructura:

```
Dictionary<string, GameObject> prefabDictionary;
```

PrefabDictionary es el nombre del diccionario, string es el tipo de llave o identificador y GameObject es el objeto que se almacenar.

```
public string name;
```

```
public string id;
```

```
public Vector3 position;
```

```
public Vector3 localScale;
```

```
public Quaternion rotation;
```

De esta manera se guarda el nombre, la posición, rotación, escala y el ID de los objetos que estén en la escena, antes de pasar la información al archivo .txt se realiza una serealización de los datos para almacenarlos y estos se guardan en un formato binario dentro del archivo txt.

Cabe aclarar que los objetos pueden tener el mismo nombre y por eso se utiliza el ID para identificarlos, el diccionario se encarga de ordenar la información y de pasarla al serealizador y en el caso de la carga simplemente se deserializa el archivo para su respectiva carga y se genera los objetos en la escena.

Los archivos .txt se almacenan en la carpeta DataSaveTour, en esa carpeta se crean los archivos correspondientes a cada Slot (ranura de almacenamiento) de la siguiente manera:

```
QuickSave1.txt
```

```
QuickSave2.txt
```

```
QuickSave3.txt
```

```
QuickSave4.txt
```

### **6.3. Infraestructura de hardware y redes**

El Framework fue desarrollado en un computador de mesa que cuenta con las siguientes características:

Sistema operativo Windows 10

Procesador Ryzen 3200G

RAM 16 GB

Tarjeta gráfica Vega 8

Las pruebas para el sistema operativo Windows se realizaron en el mismo computador y para las pruebas en Android se llevaron a cabo en tres dispositivos diferentes dos de gama media y uno de gama alta

El software está disponible sistemas operativos Windows 10 y Android 7.0 para poder ejecutar el Framework en Windows se necesita como requisito mínimo las siguientes Especificaciones:

Para computadoras son recomendable los siguientes requisitos de hardware:

Tarjeta gráfica:

NVIDIA GTX 1050 Ti/AMD Radeon RX 470 o superior

Tarjeta gráfica alternativa

NVIDIA GTX 960/AMD Radeon R9 290 o superior

CPU: Intel i3-6100/AMD Ryzen 3 1200, FX4350 o superior

Memoria: 8 GB de RAM o más

SO: Windows 10

Para Android son los siguientes requisitos de hardware:

Pantalla de entre 4,7 y seis pulgadas de tamaño.

Resolución de al menos 1.920 x 1.080 píxeles, aunque se recomienda resolución Quad HD (2.560 x 1.440 píxeles).

Tasa de refresco de al menos 60 Hz en el modo de realidad virtual.

Soporte de Bluetooth 4.2 y Bluetooth LE.

Debe tener giroscopio, acelerómetro y magnetómetro.

4 GB de RAM.

Se necesita un visor si desea utilizar la función de ver el Tour en realidad virtual.

## 7. Pruebas del software

### 7.1. Pruebas del software

Para verificar si el prototipo del Framework cumple o no los requisitos funcionales y no funcionales se le pide a un programador externo al proyecto que probará la aplicación y analizará solamente los requisitos planteados, posteriormente el daba su opinión y llenaba una tabla con un “Si” si cumplía el requisito o un “No” si el software no lo cumple.

A continuación, observaremos las tablas:

Tabla 20.

*Verificación de los requisitos funcionales*

Requisito funcional No.	Requisito	Cumple	No cumple
RQF 1	La aplicación permitirá que el usuario pueda importar diseños de renderizado en 3D o fotos 360°.	Si	
RQF 2	La aplicación contará con diseños predeterminados de objetos 3D.	Si	
RQF 3	El usuario podrá elegir un diseño predeterminado.	Si	
RQF 4	El usuario podrá crear sus escenas.	Si	
RQF 5	La aplicación permitirá editar las escenas hechas por el usuario.	Si	
RQF 6	El usuario podrá reproducir un diseño de escena o tour hecho por él.	Si	

*Nota:* (autoría propia).

Tabla 21.

*Verificación de los requisitos no funcionales*

Requisito no funcional No.	Requisito	Cumple	No cumple
RQNF 1	La aplicación será ejecutable para dispositivos Android y Windows.	Si	
RQNF 2	La aplicación se podrá ejecutar desde dispositivos Android desde la versión 7.0 o superior.	Si	
RQNF 3	La aplicación podrá ejecutarse en dispositivos con 4GB de RAM o superior.	Si	
RQNF 4	La aplicación podrá ejecutarse sin conexión a internet.	Si	
RQNF 5	La aplicación solo permite importar renderizado de formato OBJ junto con su respectivo MTL	Si	
RQNF 6	La aplicación solo permite importar fotos de formato JPG o PNG	Si	

*Nota:* (autoría propia).

## 7.2. Pruebas de usabilidad

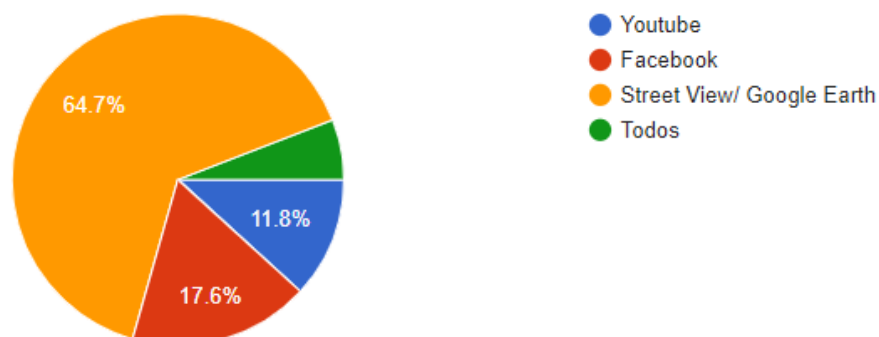
Para las pruebas de usabilidad se realizó una encuesta a 17 personas, se trató en lo posible que las personas encuestadas encajaran con el perfil de usuario al igual se les preguntó qué tan frecuente consumen contenido en formato de Realidad Virtual o video e imagen 360°.

Al finalizar se analizaron los siguientes gráficos de tortas:



¿En que plataformas consumes contenido en Realidad Virtual o imágenes en 360°?

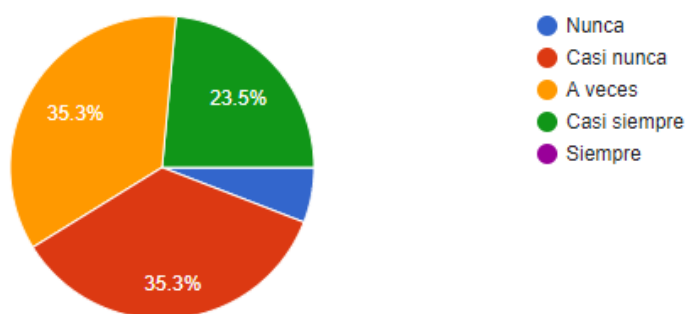
17 respuestas



**Figura 30.** Gráfico de torta pregunta 1 de la encuesta prueba de usabilidad del Framework. Autoría propia.

¿Qué tan frecuente consume ese tipo de contenido? (Realidad Virtual o videos/imágenes 360°)

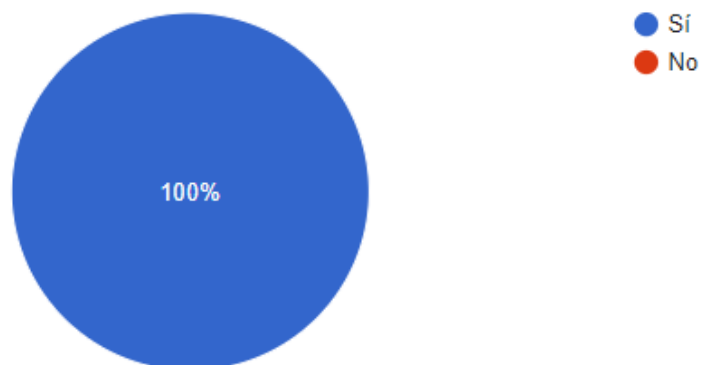
17 respuestas



**Figura 31.** Gráfico de torta pregunta 2 de la encuesta prueba de usabilidad del Framework. Autoría propia.

¿Usted logra ver claramente los botones de la interfaz?

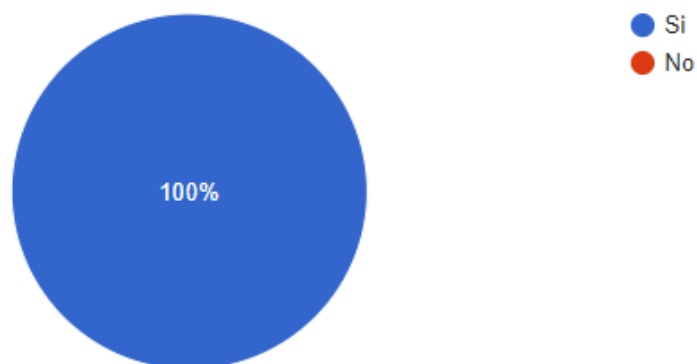
17 respuestas



**Figura 32.** Gráfico de torta pregunta 3 de la encuesta prueba de usabilidad del Framework. Autoría propia.

¿Es intuitivo el uso de la aplicación?

17 respuestas

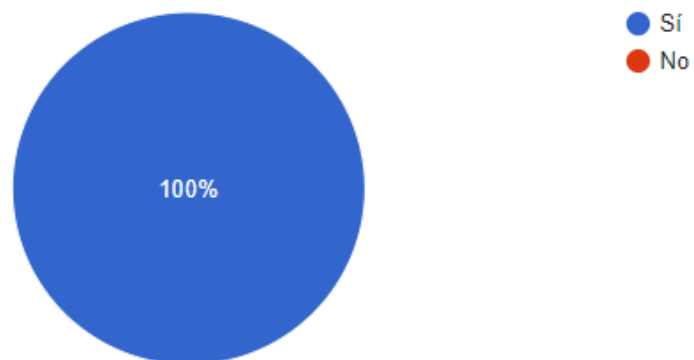


**Figura 33.** Gráfico de torta pregunta 4 de la encuesta prueba de usabilidad del Framework. Autoría propia.

---

¿Considera que el entorno de la aplicación es amigable?

17 respuestas



---

Si su respuesta fue No escriba el porque

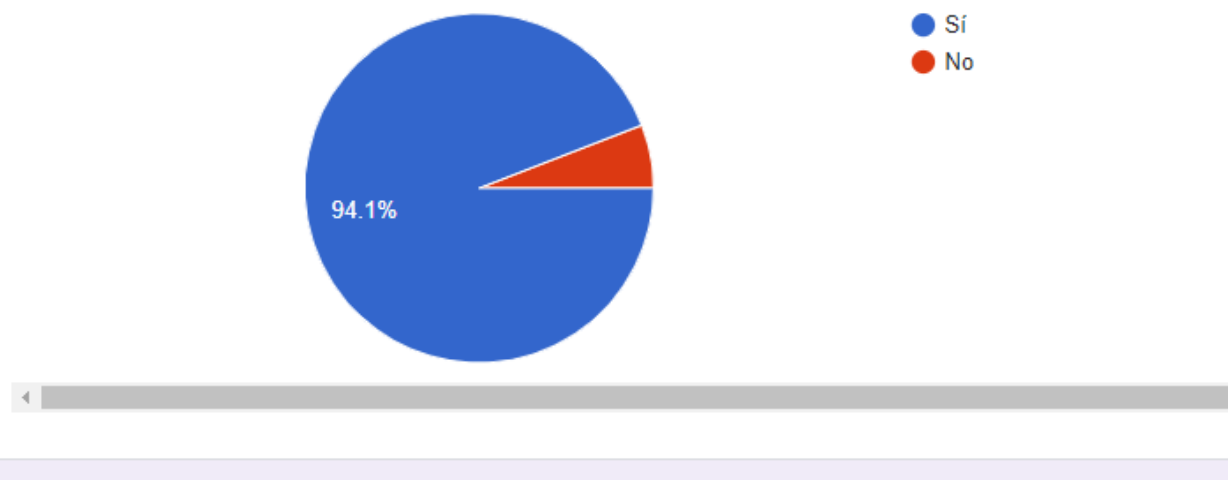
0 respuestas

Todavía no hay respuestas para esta pregunta.

**Figura 34.** Gráfico de torta pregunta 5 y pregunta 6 de la encuesta prueba de usabilidad del Framework. Autoría propia.

¿Fue fácil para usted importar un objeto 3D o imagen en la escena?

17 respuestas



Si su respuesta fue No escriba el porque

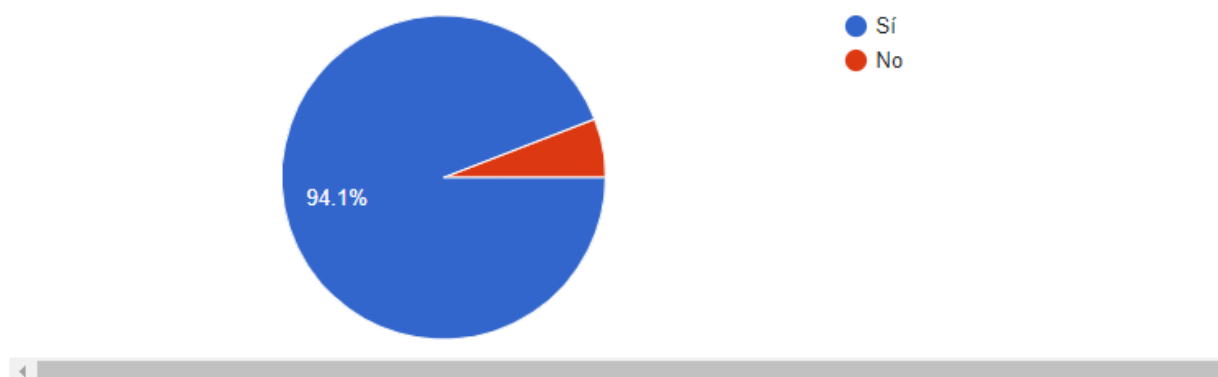
1 respuesta

Porque jamás e tenido la necesidad de hacerlo

**Figura 35.** Gráfico de torta pregunta 7 y 8 de la encuesta prueba de usabilidad del Framework.  
Autoría propia.

¿Fue fácil para usted colocar los puntos de control u objetos 3D en la escena?

17 respuestas



Si su respuesta fue No escriba el porque

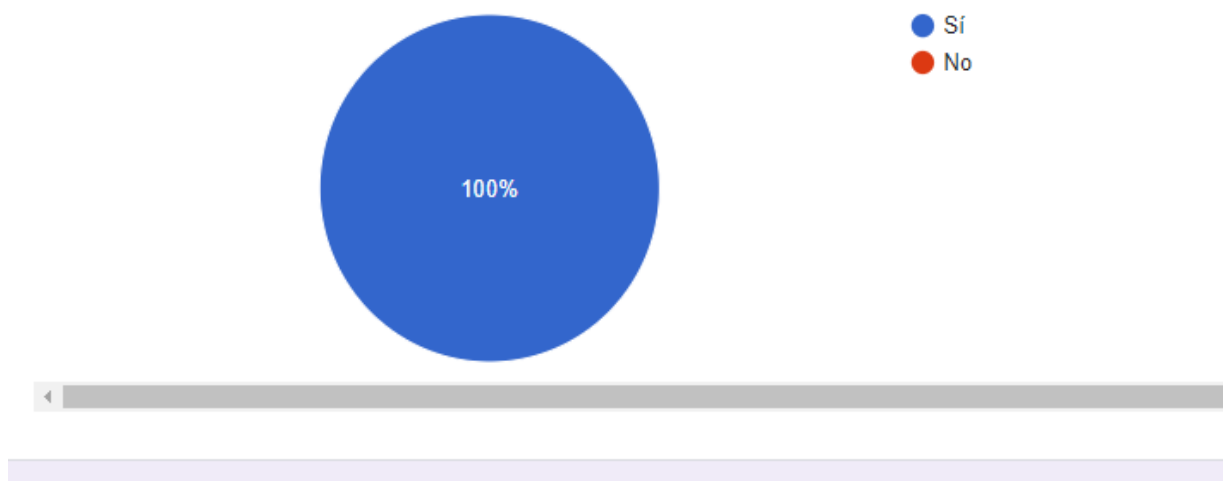
1 respuesta

No e tenido la necesidad

**Figura 36.** Gráfico de torta pregunta 9 y 10 de la encuesta prueba de usabilidad del Framework.  
Autoría propia.

¿Fue fácil para usted reproducir el recorrido?

17 respuestas



Si su respuesta fue No escriba el porque

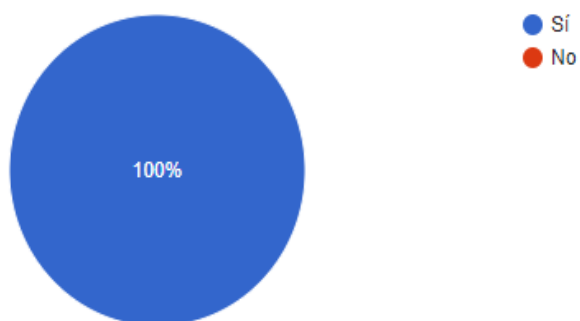
0 respuestas

Todavía no hay respuestas para esta pregunta.

**Figura 37.** Gráfico de torta pregunta 11 y 12 de la encuesta prueba de usabilidad del Framework. Autoría propia.

¿Cree usted que esta aplicación sirve para el desarrollo de creación y consumo de contenido en Realidad Virtual o 360°?

17 respuestas



**Figura 38.** Gráfico de torta pregunta 13 de la encuesta prueba de usabilidad del Framework. Autoría propia.

¿Qué funcionalidad le gustaría agregar en la próxima versión?

6 respuestas



**Figura 39.** Gráfico de torta pregunta 14 de la encuesta prueba de usabilidad del Framework. Autoría propia.

¿Qué mejoraría en las funciones o interfaz (visual/funcional) existentes de la aplicación?

6 respuestas



**Figura 40.** Gráfico de torta pregunta 15 de la encuesta prueba de usabilidad del Framework. Autoría propia.

Como conclusión se determinó que la aplicación cumple con las expectativas de usabilidad logrando un 94.1% de aceptación para las funcionalidades y un 100% para la interfaz, la encuesta también sirvió para identificar qué mejoras serían óptimas para una futura versión del prototipo.

## **Conclusiones**

En un inicio se pensó que el Framework sería de utilidad solo para la aplicación a la que se enlazara pero después del desarrollo, las pruebas y varias opiniones se llegó a la conclusión de que este prototipo es el inicio de una nueva forma de generar contenido para Realidad Virtual enfocado al nicho de recorridos o entornos ya sea para ventas, muestras de construcciones o instalaciones, en un futuro se podría convertir a un entorno de edición más interactivo captando el movimiento de las manos con controles mediante el Oculus Rift o dispositivos similares llevando al prototipo a una inmersión más lograda, también se podría implementar un apartado específico para los Smartphone como lo sería la Realidad Aumentada mostrando los objetos en 3D de la escena o el mismo recorrido construido.

Esta aplicación podría abrir un nuevo mercado si es que aún no lo hay de manera extensa dado el caso de que ya exista convertirá ese servicio de creación de tours para Realidad Virtual en algo más asequible gracias a los Smartphone mejorando la competencia, el cómo se ofrece el servicio, su acabado final y los costos.



### **Recomendaciones**

Como recomendaciones se identificó que en el caso del Framework implementado en dispositivos móviles con sistema operativo Android es de gran dificultad escribir datos en la memoria interna del dispositivo lo que seguirá siendo un inconveniente en futuras versiones del prototipo.

El software podría seguir por una línea de desarrollo enfocada a aplicar la experiencia de realidad virtual en el modo de edición y optimizar las funciones de guardar/cargar e importar.

### Referencias

- B. Labra, P. Ramírez, O. Zezzatti, O. Hernández. (2017). Simulador ROV multiplayer para escenarios petroleros submarinos. p. 67. Recuperado de: [https://www.rcs.cic.ipn.mx/2017\\_140/Simulador%20ROV%20multiplayer%20para%20escenarios%20petroleros%20submarinos.pdf](https://www.rcs.cic.ipn.mx/2017_140/Simulador%20ROV%20multiplayer%20para%20escenarios%20petroleros%20submarinos.pdf)
- Brookes, J., Warburton1, M., Alghadier, M., Williams, M., y Mushtaq, F. (2020). Studying human behavior with virtual reality: The Unity Experiment Framework. *Behav Res*45, 455(463) doi: <https://doi.org/10.3758/s13428-019-01242-0>
- Cid Rodríguez Álvaro y Martínez Zarzuela Mario. (2012). Diseñador de espacios 3D con realidad aumentada, realidad virtual y edición online del entorno (Trabajo fin de Master, Universidad de Valladolid). Recuperado de: <https://uvadoc.uva.es/bitstream/handle/10324/2709/TFM-G88.pdf?sequence=1&isAllowed=y>
- Cuervo, W., y Ballesteros, J. (2016). Framework para desarrollo de aplicaciones educativas móviles, basado en modelos de enseñanza. *Praxis y Saber*, 8(17). doi: <https://doi.org/10.19053/22160159.v8.n17.2018.7204>
- Domínguez, E. (2015). Periodismo inmersivo o como la realidad virtual y el videojuego influyen en la interfaz e interactividad del relato de actualidad. *Open Journal Systems*, 24(4). doi: <https://doi.org/10.3145/epi.2015.jul.08>
- Elfaragy, M., y Rizq, A. (2018). VIR MUF: THE VIRTUAL MUSEUM FRAMEWORK. *Scalable Computing: Practice and Experience*, 19(2). doi: <https://doi.org/10.12694/scpe.v19i2.1349>
- González, I. (2018). Market Cart App: aplicación móvil para la gestión de compra de víveres en línea. *TIA*, 6(1). Recuperado de: <https://revistas.udistrital.edu.co/index.php/tia/article/view/9687/pdf>
- Martínez, G., Camacho, G., Biancha, D. (2010). Diseño de un Framework web para el desarrollo dinámico de aplicaciones. *Scientia Et Technica*, 16(44). Recuperado de: <https://www.redalyc.org/pdf/849/84917316032.pdf>
- Martínez, P., y Perlin, K. (2011). Presente y futuro de la Tecnología de la realidad virtual. 15(5). Recuperado de: [https://d1wqtxts1xzle7.cloudfront.net/59541849/4-Realidad\\_Virtual20190605-15699-1qa1682.pdf?1559782041=&response-content-](https://d1wqtxts1xzle7.cloudfront.net/59541849/4-Realidad_Virtual20190605-15699-1qa1682.pdf?1559782041=&response-content-)

disposition=inline%3B+filename%3DFrancisco\_Javier\_Perez\_Martinez\_Presente.pdf&Expires=1605230502&Signature=A~mxdQGQ5gIEnze6O~8q8-e644zgbwWmpWsADqeS6AsC6JrPKUULTIV08EFXL191vngyioxTKim5Ktby99g0qtT4sNth52OuNZzfZ5pWSe9LPXrc2tWuICphDQKsmfTH3tRB4yXNzs4eJWomnJzvH1SuOxybNRPLu5QY5UIXz7S-Gcuh8z87tMk~CIVACsXoSrIDU370CHtCx8Neu8YvF6XyquY4tMXDGg1boyUaBISSk6uGVenCOYPBegCmyhIJd5AvEDWAq0XmkijC~p7HXYhRMId4A8aMYLeashnLo5NrG235wVUUsrXN1nFL7jVV59dRigOCRATnd012CIqeIw\_\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

- Montoya, H., y León, D. (2014). Descripción de un Framework metodológico para el desarrollo de aplicaciones relacionadas con el patrimonio cultural. *ReVisión*, 7(2). Recuperado de: <https://rua.ua.es/dspace/bitstream/10045/37972/1/148-1324-1-PB.pdf>
- Ponder, M., Papagiannakis, G., Molet, T., Magnenat, N., y Thalmann, D. (Junio, 2003). VHD ++ Development Framework: Towards Extendible, Component Based VR/AR Simulation Engine Featuring Advanced Virtual Character Technologies. *Computer Graphics International*, Tokio, Japón. Recuperado de: [file:///C:/Users/User/Downloads/VHD\\_Development\\_Framework\\_Towards\\_Extendible\\_Compo%20\(1\).pdf](file:///C:/Users/User/Downloads/VHD_Development_Framework_Towards_Extendible_Compo%20(1).pdf)
- Safadel, P., y White, D. (2020). Effectiveness of Computer-Generated Virtual Reality (VR) in Learning and Teaching Environments with Spatial Frameworks. *Applied Sciences*, 10(16). doi: <https://doi.org/10.3390/app10165438>
- Sermet, Y., y Demir, I. (2020). GeospatialVR: A Web-based Virtual Reality Framework for Collaborative Environmental Simulations. p. 2. Recuperado de: <https://eartharxiv.org/repository/view/41/>
- Somerville, I. (2011). *Ingeniería de Software [e-book]*. Recuperado de: [http://artemisa.unicauca.edu.co/~cardila/Libro\\_Sommerville\\_9.pdf](http://artemisa.unicauca.edu.co/~cardila/Libro_Sommerville_9.pdf)
- Torres, E. (2019) Un museo para un arte sin límites: mori building digital art museum. *Diferents. Revista De Museus*, 1(4). doi: <http://dx.doi.org/10.6035/Diferents.2019.4.5>
- Wang, Y., Ijaz, K., y Calvo, R. (diciembre, 2017). A Software Application Framework for Developing Immersive Virtual Reality Experiences in Health Domain. *Congress*

International, Sídney, NSW, Australia. Recuperado de:  
<https://rafaelacalvo.files.wordpress.com/2013/11/17-lsc-tifan-final.pdf>

Wessels, S., Ruther, H., Bhurtha, R., y Schroeder, R. (2014). Design and creation of a 3D virtual tour of the world heritage site of Petra, Jordan. p. 4. Recuperado de:  
[http://africageoproceedings.org.za/wp-content/uploads/2014/06/97\\_Wessels\\_Ruther\\_Bhurta\\_Schroeder.pdf](http://africageoproceedings.org.za/wp-content/uploads/2014/06/97_Wessels_Ruther_Bhurta_Schroeder.pdf)